Effective Symbolic Regression-Based Petrophysical Model Development Workflow and Efficient Software Tool for Enabling Easy Utilization

Wei Shao[1,*], *Songhua* Chen[1]
Hyung T. Kwak[2], *Jun* Gao[2], *Mohammed* Abdul-Qadir[2], *Abdullah* Alkhaldi[2], and *Gabor* Hursan[2]

[1]Halliburton, USA
[2]Saudi Aramco, Kingdom of Saudi Arabia

Abstract. Developing petrophysical interpretation models integrating multiphysics measurements is challenging. Symbolic regression (SR) offers a key advantage over other machine learning methods by generating analytical expressions enabling petrophysicists to assess consistency with physical principles and understand relevancy of the input variables. Existing SR packages are not optimized for petrophysical modeling, which often relies on limited and error-prone core samples. To address this, we developed a software solution using genetic programming-based SR to produce transparent, interpretable prediction equations. Our workflow consists of five components: (1) statistical feature selection, (2) symbolic regression for multiphysics fusion, (3) ensemble modeling, (4) conditional branching for formation heterogeneity, and (5) model discrimination for optimization. To enhance usability, we integrated multiple open-source SR packages into a user-friendly interface, providing tools for data preparation, visualization, and model evaluation. This implementation simplifies workflow execution, making SR-based petrophysical modeling accessible to users without advanced programming skills.

1 Introduction

Developing accurate and reliable petrophysical interpretation models remains a critical challenge in subsurface reservoir characterization. The integration of multiphysics measurements, such as well logs, core samples, and data from Routine Core Analysis (RCA) and Special Core Analysis (SCA), is often essential for constructing robust models that provide meaningful insights into reservoir properties. However, the complexity and inherent uncertainties in these datasets make it difficult to establish empirical relationships that align with physical principles while maintaining prediction accuracy.

Symbolic regression (SR) [1, 2] has emerged as a powerful machine learning approach for developing predictive models in petrophysics. Unlike traditional machine learning models such as neural network models, which often act as black boxes, SR generates explicit analytical expressions that describe relationships between input parameters and target properties. This transparency enables petrophysicists to assess the consistency of derived equations with known physical laws, ensuring model interpretability and allowing for a deeper understanding of variable importance in predictions.

Despite the availability of commercial and open-source SR packages, most existing solutions are not optimized for petrophysical applications, particularly when dealing with limited and noisy core sample data. Measurement errors, formation heterogeneity, and nonlinearity in petrophysical properties further complicate the development of reliable models. The absence of tailored SR tools designed specifically for petrophysical analysis has limited the broader adoption of this technique within the industry.

To address these challenges, we have developed a comprehensive workflow that utilizes genetic programming-based symbolic regression to create transparent and interpretable petrophysical models. The workflow consists of five essential components: (1) statistical methods for feature selection, (2) symbolic regression for fusing multiphysics measurements, (3) an ensemble procedure to integrate multiple SR models, (4) conditional branching to account for formation heterogeneity, and (5) a model discrimination framework to optimize and validate results.

While this workflow has demonstrated effectiveness in generating accurate petrophysical models, its adoption has been hindered by the need for significant scientific expertise in evaluating and interpreting SR outputs. Different SR packages may yield dissimilar equations and varying model performances, requiring users to make informed decisions about model selection and validation. This complexity poses a barrier to petrophysicists who may not have extensive experience with symbolic regression methodologies.

* Corresponding author: wei.shao@halliburton.com

To facilitate the practical application of our workflow, we have developed user-friendly software implementation that integrates multiple open-source SR packages. The software features an intuitive interface for data preparation, visualization, and exploration, enabling users to interact seamlessly with their datasets. In addition to executing the five key workflow components, it provides functionalities for postprocessing and evaluating SR-based petrophysical models. By simplifying model development and validation, our software aims to bridge the gap between advanced symbolic regression techniques and real-world petrophysical analysis, making it accessible to a broader range of practicing professionals.

To demonstrate the effectiveness of our approach, we apply the SR-based petrophysical interpretation workflow to two datasets: one from an unconventional reservoir and the other from carbonate formation. Specifically, we derive formation resistivity factor models for the carbonate reservoir and permeability equations for the heterogeneous unconventional reservoir. We demonstrate how each workflow component enhances model performance. This study highlights the potential of SR-based techniques in advancing petrophysical modeling and improving reservoir characterization in complex geological settings.

2 Overview of existing petrophysical modelling techniques

Petrophysical modeling for new reservoirs typically involves a range of approaches, each with its strengths and limitations. A common method is to adapt an existing model by tuning its parameters, often in combination with facies-based techniques for handling complex reservoirs. Facies identification, however, relies on subjective methods such as petrographic analysis and core data, which may not be applicable to heterogeneous formations. Furthermore, discrete facies models often struggle to capture continuous or transitional variations within formations, limiting their effectiveness in complex geological settings.

Another approach is mechanistic modeling, which is based on fundamental physical principles. While these models are grounded in physics, they often require simplifications to make the equations solvable, reducing their accuracy when applied to complex systems with heterogeneous characteristics.

Empirical models offer a data-driven alternative, relying on observed relationships between petrophysical parameters. However, these models can overlook important physical constraints and are often limited by the diversity of training data and the simplistic functional forms they use, making them less suitable for capturing the complexity of natural reservoirs.

Artificial Intelligence (AI)-based models have gained traction as a powerful tool for capturing nonlinear relationships among petrophysical properties. These models can potentially offer higher accuracy by recognizing complex patterns within the data. However, they often operate as "black boxes," offering limited transparency regarding how predictions are made and whether they adhere to physical or geological principles. This lack of interpretability—especially when integrating diverse measurements like NMR, resistivity, and acoustic logs—can limit trust in their results and hinder their adoption in operational workflows.

Furthermore, AI-based models typically require large datasets to effectively train the model. However, many petrophysical models rely on core samples, and the number of available core samples is often limited due to cost and time constraints. This lack of sufficient data can hinder the performance of AI models, as they may struggle to generalize without a robust dataset to learn from.

3 Symbolic Regression for petrophysical modelling

Unlike traditional machine learning, Symbolic Regression produces explicit mathematical equations that petrophysicists can examine, validate, and interpret. This transparency enhances model evaluation by ensuring physical plausibility and enabling targeted adjustments.

In formation evaluation, multiple logging tools are often used to estimate the same petrophysical parameter, but integrating these multi-sensor outputs into a unified model is challenging. SR provides an efficient and interpretable solution by combining data from various sources into a coherent mathematical framework.

We have applied SR to a range of petrophysical and reservoir description problems, as demonstrated in [3] and [4], which showcase its strengths in interpretability and model flexibility. To develop petrophysical models, we evaluated several SR frameworks, including four open-source algorithms—PySR [5], Rils-Rols [6], HeuristicLab Genetic Regressor [7], and AI-Feynman [8]—as well as one commercial tool, DataRobot's Eureqa [9]. Each method has unique strengths; combining different SR approaches can further enhance prediction performance.

Some symbolic regression (SR) frameworks may be less suited for petrophysical modeling. For example, while AI-Feynman performs well on purely physics-based problems by leveraging principles such as symmetry, dimensional reduction, and variable separability, it may be less effective for complex, heterogeneous rock characterization applications. Based on our experience, we have applied AI-Feynman to several petrophysical problems with limited success.

PySR, HeuristicLab Genetic Regressor, and DataRobot's Eureqa are symbolic regression techniques based on genetic programming (GP). In contrast, Rils-Rols utilizes iterated local search and ordinary least squares methods. However, it supports only a limited and predefined set of operators and notably lacks branching operators, which are essential for capturing the heterogeneity of complex reservoirs. Even so, we achieved some success with it on certain petrophysical problems.

HeuristicLab is a GUI-based genetic programming (GP) platform, primarily designed for experimentation with the genetic process, including selection, crossover, mutation, parameter tuning, and fitness evolution. While

it offers considerable flexibility for algorithmic exploration, its graphical interface does not adequately meet the practical needs of petrophysicists. Additionally, we experienced only limited success when applying it to our petrophysical problems.

DataRobot's Eureqa is web-based commercial software. Its graphical user interface is more focused on model tuning, evaluation, and interpretability. However, it still lacks several features required for petrophysical applications. Despite these limitations, we have successfully developed various petrophysical models using Eureqa ([3], [4]).

PySR is a code-based symbolic regression (SR) software library that requires advanced programming skills to be used effectively. We have found PySR to be very powerful for handling small- to medium-sized petrophysical datasets. However, it can be prone to overfitting, often producing overly complex models that are neither interpretable nor explainable.

To address these challenges and enable the efficient application of SR to petrophysical problems, we developed a GUI-based software platform that integrates several SR packages. This platform includes GUI features specifically tailored to the needs of petrophysicists—such as tools for data evaluation, quality control, and model validation—making these advanced techniques more accessible to users without programming backgrounds.

Moreover, simply applying SR to develop petrophysical models does not always yield optimal results, largely due to the inherent noise present in well logging and core analysis data. While SR is generally well-suited for small datasets, core data is often extremely limited. For example, only 29 core samples were available for constructing the permeability model discussed in the Examples section. To overcome these limitations, we developed a comprehensive workflow specifically designed for complex petrophysical modeling tasks, which are detailed in the following section.

4 SR workflow for petrophysical modelling

The workflow comprises five key components, outlined below. A detailed description can be found in the paper by Chen et al. [3], while a summary is provided here.

4.1 Correlation heatmap for multiphysics measurement selection

The first step in developing SR-based petrophysical models is selecting relevant measurement data as input variables. While physics provides initial guidance, rock heterogeneity can obscure clear dependencies. We use correlation heatmaps to gain an initial understanding of variable relationships, guide feature selection and engineering, and inform the evaluation of the models.

4.2 Symbolic Regression (SR)

Our primary focus is on genetic programming-based Symbolic Regression (SR) to derive mathematical expressions for petrophysical problems. This approach searches the mathematical expression space using a genetic algorithm, evolving equations through operations such as crossover and mutation. Its stochastic nature promotes the generation of diverse and valid solutions, making it both robust and adaptable for integrating a wide range of measurements.

Our software can also integrate non–genetic programming-based SR packages, such as Rils-Rols. However, in this paper, we primarily focus on the integration of PySR as an example.

4.3 Ensemble modeling

To enhance model reliability, ensemble modeling aggregates multiple SR-generated equations from a single SR package or across multiple packages. This approach ensures generalizability, particularly when data size is limited. The ensemble model is formed using weighted combinations of base models, improving predictive accuracy while maintaining interpretability.

4.4 Conditional branching

Input variables often correlate with the target in distinct clusters. Traditionally, identifying such clusters relies on expert judgment, which can introduce bias. Data-driven techniques like clustering and pair plotting can help, but the resulting branches are often difficult to express mathematically and may miss optimal, non-linear splits that symbolic regression (SR) can uncover during training.

In genetic programming-based SR, branching can be controlled by including or excluding operators like max and min. While branching can enhance model accuracy, excessive use may lead to overfitting and reduce the model's ability to generalize. Therefore, branching is selectively enabled—either when unbranched models underperform or when justified by statistical correlation analysis, as illustrated in the Examples section.

4.5 Model selection criteria

Optimal model selection aims to balance three key factors: the complexity of the model's mathematical expression, the complexity of physical measurements required, and the model's prediction accuracy. Each model is evaluated and assigned an overall score based on these criteria.

Let $Loss$ represent the model's prediction error, such as mean square error (MSE). Let $C1$ denote the complexity of the model's mathematical expression, and $C2$ represent the complexity of physical measurements. The overall score of a model is defined as follows:

$$Score = Loss + \lambda * Complexity \quad (1)$$

where λ is a weighting factor that controls the trade-off between prediction error and model complexity. The

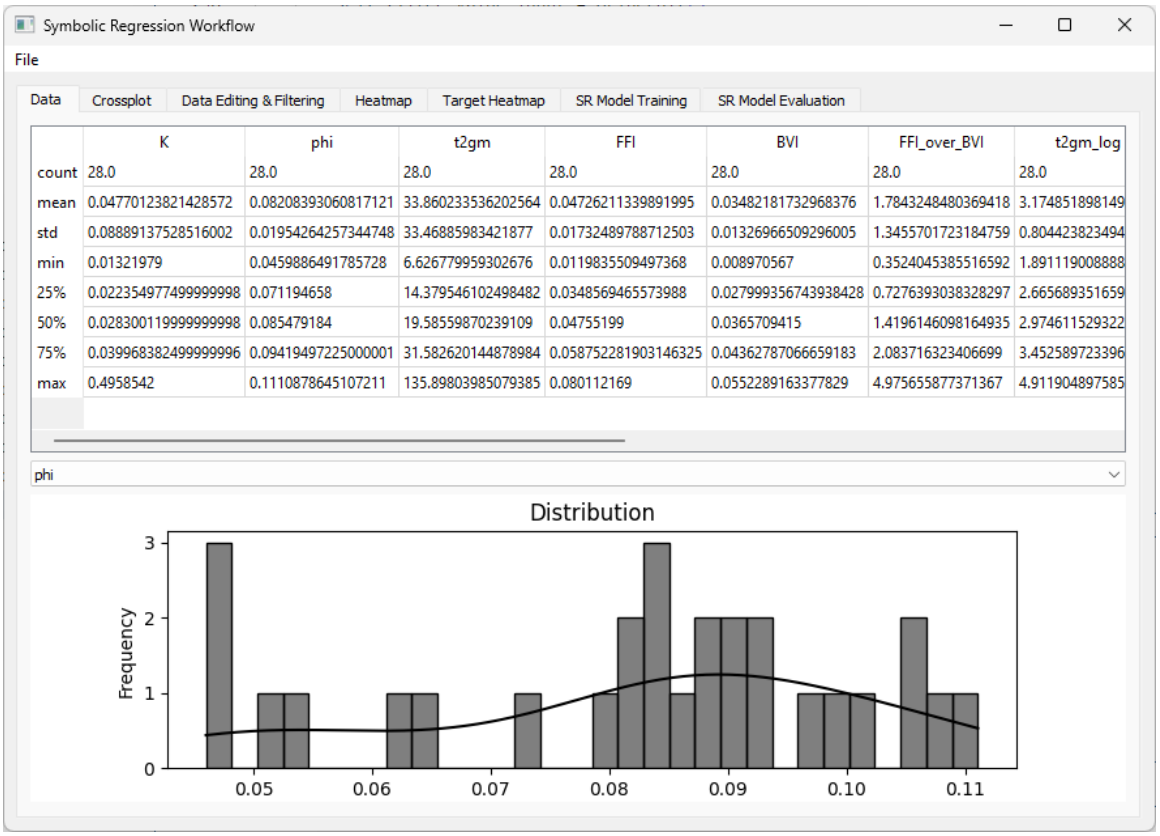
$Complexicty$ term combines the two components, $C1$ and $C2$, and is defined as:

$$Complexity = \log(C2) * C1 \qquad (2)$$

$C2$ is determined by several factors, including the sensitivity of the physical parameters, the reliability of the measured data, and the robustness of the measurements to noise. The simplest approach is to define $C2$ as the number of physical measurements used in the model, plus 1. The addition of 1 ensures that the logarithmic term remains valid, even when only a single physical measurement is included.

The software is a standalone desktop application developed using Python and the PyQt5 GUI framework, enabling the creation of feature-rich applications with a native and intuitive user interface. A desktop-based architecture is chosen over a web-based approach to allow users to develop petrophysical models efficiently with limited resources, such as a standard laptop. Given that most petrophysical models rely on a limited number of data samples, symbolic regression (SR) model development can be easily performed on modern laptops—such as those equipped with an Intel® Xeon® W-11955M CPU @ 2.60 GHz and 32 GB RAM—without the need for high-performance computing infrastructure.

5 Software implementation



	K	phi	t2gm	FFI	BVI	FFI_over_BVI	t2gm_log
count	28.0	28.0	28.0	28.0	28.0	28.0	28.0
mean	0.04770123821428572	0.08208393060817121	33.860233536202564	0.04726211339891995	0.03482181732968376	1.7843248480369418	3.174851898149
std	0.08889137528516002	0.01954264257344748	33.46885983421877	0.01732489788712503	0.01326966509296005	1.3455701723184759	0.804423823494
min	0.01321979	0.0459886491785728	6.626779959302676	0.0119835509497368	0.008970567	0.3524045385516592	1.891119008888
25%	0.022354977499999998	0.071194658	14.379546102498482	0.0348569465573988	0.027999356743938428	0.7276393038328297	2.665689351659
50%	0.028300119999999998	0.085479184	19.58559870239109	0.04755199	0.0365709415	1.4196146098164935	2.974611529322
75%	0.039968382499999996	0.09419497225000001	31.582620144878984	0.058752281903146325	0.04362787066659183	2.083716323406699	3.452589723396
max	0.4958542	0.1110878645107211	135.89803985079385	0.080112169	0.0552289163377829	4.975655877371367	4.911904897585

Fig. 1. Symbolic regression workflow software graphic user interface

Figure 1 shows a snapshot of the user interface, which is organized into **seven panels** (each corresponding to a tab in Figure 1) designed to support the execution of the five key components of the SR workflow. These panels include:

1. Data Panel
2. Crossplot Panel
3. Data Editing and Filtering panel,
4. Heatmap Panel,
5. Target Heatmap panel,
6. SR Model Training Panel,
7. SR Model Evaluation Panel

Each panel is designed to support specific tasks in the petrophysical modeling workflow, including data exploration, feature selection, model training, and performance evaluation. The interface incorporates a range of graphical tools for data visualization and analysis, with functionalities aligned to the typical requirements of petrophysical interpretation.

The **Data** panel provides an overview of the statistical properties of the dataset, including histograms for each feature. This allows users to gain a preliminary

understanding of data distributions and helps identify potential outliers.

The **Crossplot** panel is specifically designed for petrophysicists to identify patterns related to lithology, fluid type, porosity, and saturation. It helps users visually explore relationships between variables and select relevant features for training SR models. Additionally, it helps detect potential outliers or anomalous data samples that may affect model performance. Figure 2 illustrates an example of anomalous data in the top left corner of a density versus neutron crossplot.

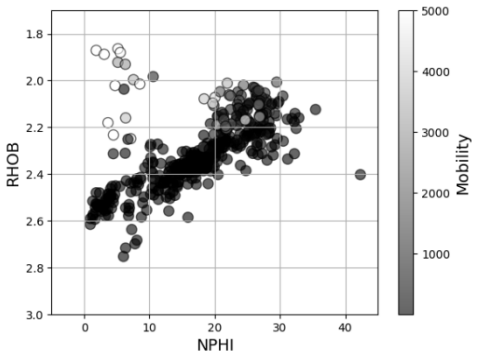


Fig. 2. Density (RHOB) versus neutron (NPHI) crossplot showing anomalous data in the top left corner, shaded according to mobility values

The two **heatmap** panels implement the first component of the SR workflow. The **Target Heatmap** panel allows users to examine the correlations between a selected feature and all other features in the dataset—not just the designated target variable for the SR model. This helps users better understand feature interdependencies and guides effective input selection for model development.

The **SR Model Training** panel implements both the **Symbolic Regression** and **Conditional Branching** components of the workflow. It provides users with the flexibility to experiment with various combinations of input features to identify those that are most relevant and consistent with established petrophysical principles. Conditional Branching is implemented using max and min operators, allowing the model to adapt to formation heterogeneity and capture different petrophysical regimes within the dataset.

Another unique feature of this panel is the ability to train the SR model using target values in either **linear** or **logarithmic** scale. This is particularly important for petrophysical parameters that span several orders of magnitude—such as permeability and mobility—which are often more appropriately analyzed and interpreted on a log scale. This flexibility enhances model stability and interpretability for such wide-ranging datasets.

The **SR Model Evaluation** panel implements the **Ensemble Modeling** and **Model Selection Criteria** components of the workflow. This panel allows users to assess and compare multiple SR models generated during training. Currently, the **Model Selection Criteria** are based on two key factors provided by the PySR framework: **model complexity** and **fitting error**. This enables users to strike a balance between accuracy and interpretability when selecting the most appropriate model for petrophysical analysis. The **Ensemble Modeling** approach is implemented through a **weighted linear combination** of the selected models at this stage, enhancing robustness and generalization of the final prediction.

5 Examples

Two examples are presented to demonstrate the process of developing petrophysical models using the proposed workflow and software.

5.1 SR-based permeability models

A total of 29 core samples were collected from five wells in an unconventional reservoir. The dataset includes porosity and permeability (P&P) measurements, nuclear magnetic resonance (NMR) data, as well as X-ray diffraction (XRD) and X-ray fluorescence (XRF) analyses. The permeability (K) of these samples is very low, accompanied by relatively low total porosities (ϕ_{total}) as illustrated in Figures 3 and 4.

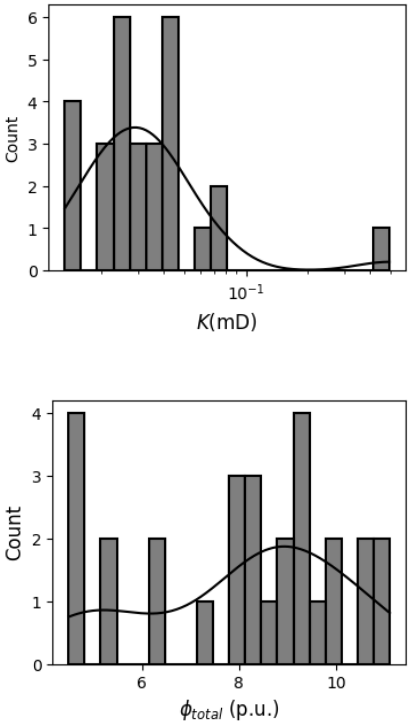


Fig. 3. Permeability and porosity distributions of the data samples.

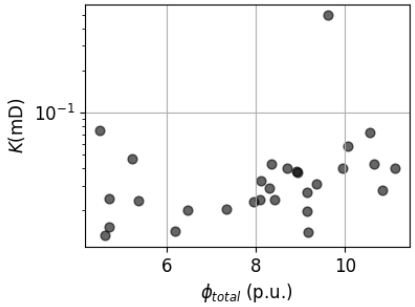


Fig. 4. Crossplot of permeability and porosity of the data samples.

Figure 5 illustrates the poor performance of traditional NMR-based models (Coates and SDR [10]) when using default parameter values. To improve their accuracy, it is common practice to calibrate the model parameters to better fit the data, as shown in Figure 6. However, even with adjusted parameters, both models still demonstrate unsatisfactory performance.

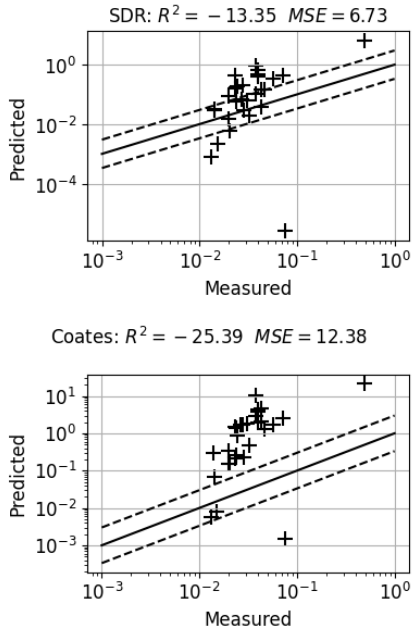
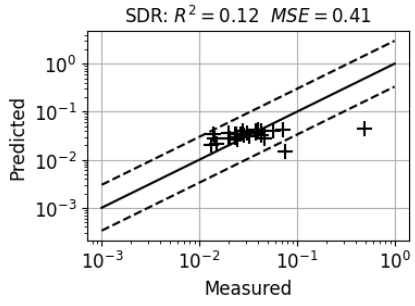


Fig. 5. SDR and Coates model performances with default parameters



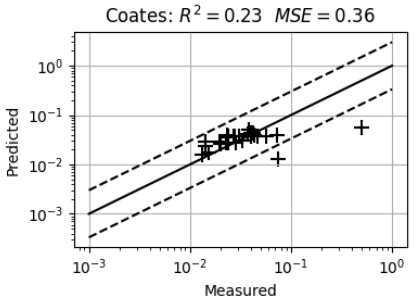


Fig. 6. SDR and Coates model performances with parameter values obtained through data fitting

The following outlines the steps of using the SR workflow and software to develop permeability models from 29 data samples obtained from unconventional reservoirs.

First, by examining the porosity distribution in Figure 3 and the crossplot of permeability versus porosity in Figure 4, several potential outliers are identified.

Using the crossplot panel in the software to further examine correlations among various features in the dataset, the outliers were narrowed down to two samples, as shown in Figure 7. One sample had $T_{2,GM}$ (geometric means of NMR transverse relaxation time) value less than 1 ms, while the other had a permeability greater than 0.1 mD.

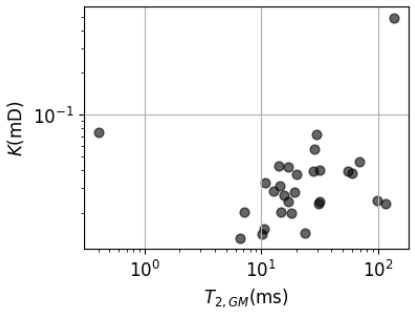


Fig. 7. Crossplot of permeability and $T_{2,GM}$ of the data samples.

Figure 8 further confirms that the sample with a $T_{2,GM}$ value less than 1 ms could be an outlier, while the other sample may not be, as it follows the general trend in terms of ϕ_{FFI} (Free Fluid Index) versus K (permeability).

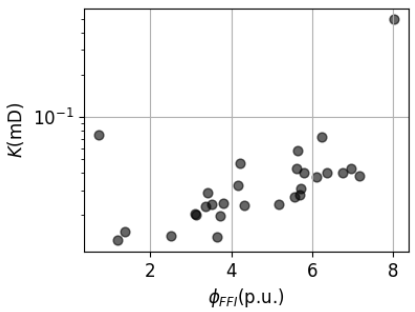


Fig. 8. Crossplot of permeability and ϕ_{FFI} of the data samples.

The identified outlier was removed using the "Data Editing and Filtering" panel, leaving 28 samples for developing the permeability model.

Before using the software to derive SR based permeability models, we re-evaluated the performance of the SDR and Coates models with the remaining 28 samples. As shown in Figure 9, the performance of both traditional permeability models improved significantly. These improved performances are used as the baseline for evaluating the SR-based permeability models.

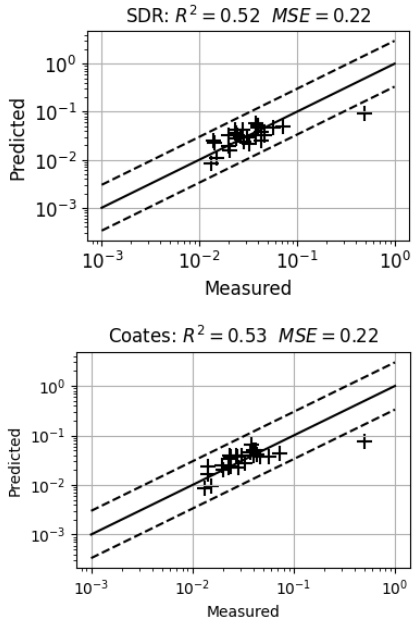


Fig. 9. SDR and Coates model performances with parameter values obtained through data fitting using the remaining 28 samples

Figure 10 shows the correlations of input variables $(T_{2,GM}, \phi_{FFI}, \phi_{total}, \phi_{BVI})$ to the target variable K using Pearson and Spearman heatmaps. Here, ϕ_{BVI} represents bound volume irreducible.

We also included several engineered input variables, such as, $\log(T_{2,GM})$, $\log(\phi_{total})$, based on our domain knowledge and experience. The SR algorithm will determine whether these engineered variables are included in the final models.

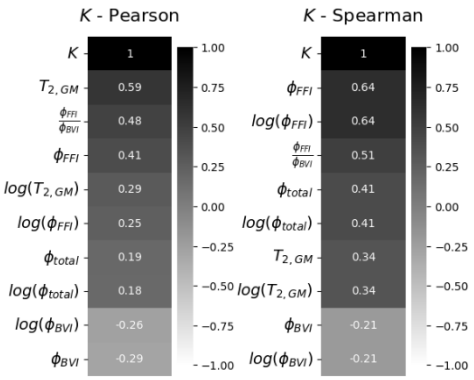


Fig. 10. Heatmaps of the input variables versus the target variable

Equation 3 presents the model derived using DataRobot's Eureqa. Figure 11 illustrates the model's performance evaluated on the entire dataset, while Figure 12 displays the performance separately for the training datasets. The results indicate that the model outperforms both the Coates and SDR models. The performance results on the testing datasets are not shown due to the small size of the testing set, as the statistical significance of the performance on these datasets is limited. Instead, the overall model performance on both the training and testing datasets is presented in Figure 11.

$$K = 0.01 + 0.12 * \phi^2 * T_{2,GM} \quad (3)$$

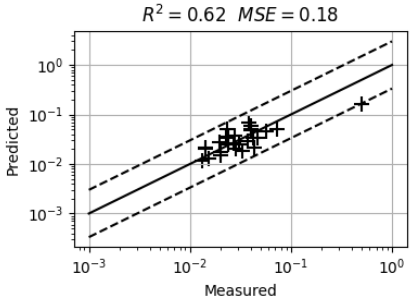


Fig. 11. Performance of the SR-based permeability model in Equation 3 evaluated over the entire dataset, encompassing both training and testing dataset

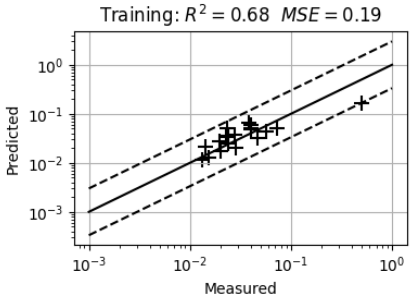


Fig. 12. Performance of the SR-based permeability model in Equation 3 evaluated over the training dataset

Equation 4 represents the model derived using the PySR. As shown in Figure 13, it demonstrates better performance on both the training and the entire datasets compared to the model in Equation 3.

However, Equation 4 is significantly more complex than Equation 3. Additionally, it shows a positive correlation between ϕ_{BVI} and K, which contradicts both the heatmaps in Figure 10 and our current understanding of the underlying relationship between K and ϕ_{BVI}.

$$K = 0.12\phi_{FFI}\left(\left(\phi_{BVI} * T_{2,GM}\right)^{\log(T_{2,GM})} + 4.16\right) \quad (4)$$

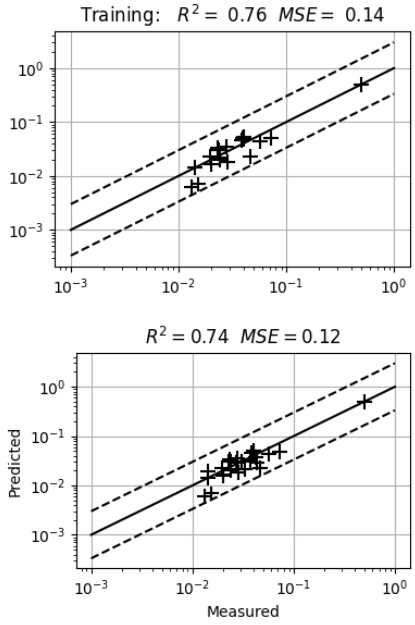


Fig. 13. Performance of the SR-based permeability model in Equation 4.

Equation 5 presents another model derived using the PySR. As shown in Figure 14, its performance has improved compared to Equation 4. Additionally, the model is simpler than Equation 4 and only slightly more complex than Equation 3.

At first glance, the term $-\frac{0.035}{1.6-\log\left(\frac{\phi_{FFI}}{\phi_{BVI}}\right)}$ may appear unusual. However, Figure 15 reveals that value 1.6 acts as a threshold, effectively separating the data point with the highest permeability from the rest of the dataset.

$$K = 0.76 * FFI * e^{-\frac{0.035}{1.6-\log\left(\frac{\phi_{FFI}}{\phi_{BVI}}\right)}} \quad (5)$$

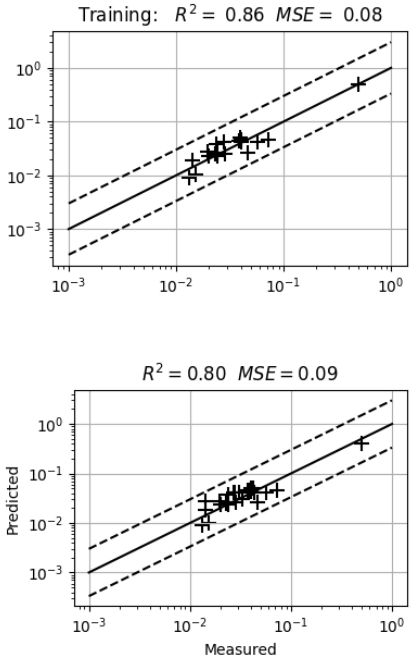


Fig. 14. Performance of the SR-based permeability model in Equation 5.

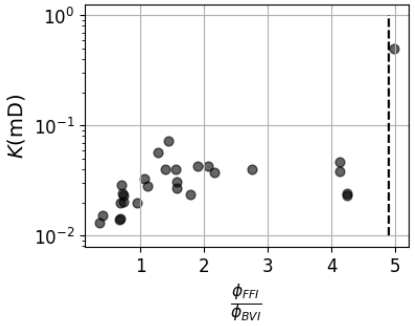


Fig. 15. Threshold value in Equation 5 used to classify the dataset into two categories

Equation 5 highlights the capability of symbolic regression (SR) methods to identify potential heterogeneity or facies within the formation. It was derived without the use of branching operators such as *max* or *min*. In contrast, Equation 6 incorporates SR branching operators and demonstrates slightly improved performance over Equation 5, as shown in Figure 16.

Equation 6 uses $T_{2,GM}$ as the branching variable, while Equation 5 relies on the ratio used $\frac{\phi_{FFI}}{\phi_{BVI}}$. Interestingly, Figure 17 reveals that both equations segment the dataset in a similar way, despite using different variables. Together, these models provide complementary insights into the underlying structure of the dataset.

$$K = \begin{cases} FFI * e^{T_{2,GM}-134.08} & if\ T_{2,GM} > 133.7420 \\ FFI * e^{-0.35} & otherwise \end{cases} \quad (6)$$

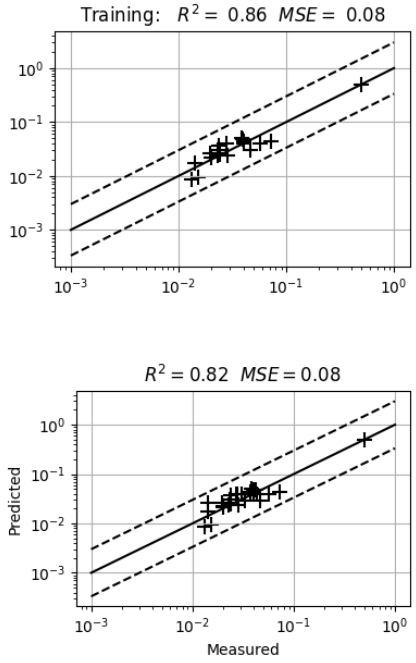


Fig. 16. Performance of the SR-based permeability model in Equation 6.

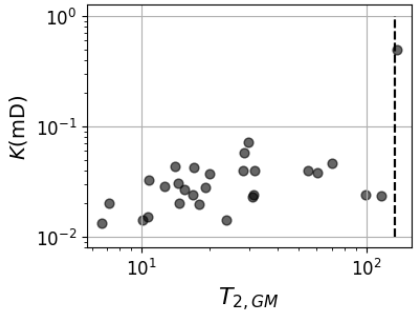


Fig. 17. Threshold value in Equation 6 used to classify the dataset into two categories

5.2 SR-based resistivity models

A total of 30 core samples from carbonate reservoirs were used to develop symbolic regression (SR)-based resistivity models. A previous study [3] detailed several SR-based models derived using the DataRobot' Eureqa. In this paper, we introduce a new resistivity factor model (Equation 7), developed using the PySR. This model aims to enhance our understanding of the reservoirs by examining and comparing the mathematical formulations of different SR-based models.

One of the key objectives is to derive the cementation factor *m*, which is traditionally treated as a constant. However, using a constant value for *m* is often insufficient in the context of complex carbonate reservoirs. A more common approach involves using facies-dependent *m* values, but identifying facies can be challenging in practice.

$$F = \frac{1}{\phi_{total}{}^{\phi_{macro}+1.9}} \tag{7}$$

Equation 7 presents the resistivity factor model derived from PySR, and Figure 18 illustrates the model's strong performance. Notably, the cementation factor can be expressed as:

$$m = \phi_{macro} + 1.9 \tag{8}$$

where ϕ_{macro} represents the macro porosity, expressed as a fraction.

Figure 19 shows the distribution of m as calculated using Equation 8, which falls within the expected ranges for carbonate reservoirs.

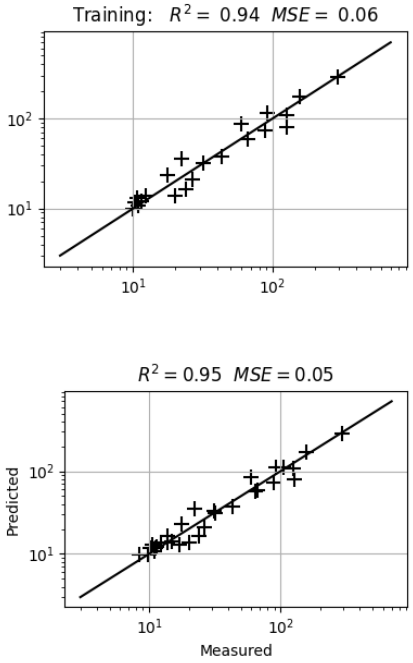


Fig. 18. Performance of the SR-based resistivity factor model in Equation 7.

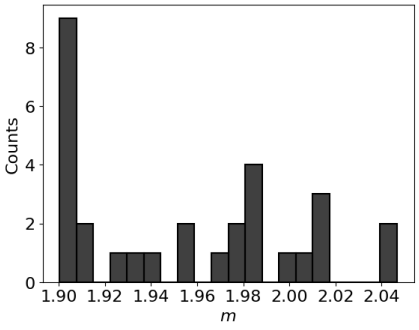


Fig. 19. *m* distributions calculated using Equation 8

Equations 9 and 10 present the resistivity models derived from DataRobot [3]. Figure 20 displays the distribution of *m*, while Figure 21 illustrates the model's performance.

$$F = \frac{1.93}{{\phi_{total}}^{m}} \quad (9)$$

$$m = 0.21\,log(T_{2,GM}) - 0.29log\ (\phi_{total}) \quad (10)$$

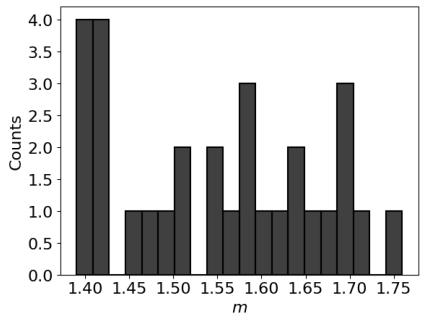


Fig. 20. m distributions calculated using Equation 10

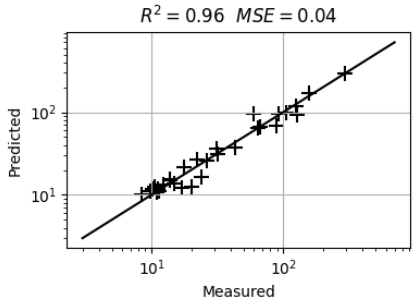


Fig. 21. Performance of the SR-based resistivity factor model in Equation 9

Compared to the models from PySR (Equations 7 and 8), the overall performance is similar. However, the distributions of the cementation factor m differ noticeably (Figures 19 and 20). These differences are discussed from the following perspectives.

First, the comparison between the two different cementation equations provides deeper insight into the lithology of these rocks than the standard Archie equation [11]. Equation (9) indicates that both pore size and porosity influence the cementation factor. In this rock set, although porosity and overall pore size (represented by $T_{2,GM}$) are positively correlated, as shown in Figure 22, their individual relationships with the cementation factor differ as shown in Figures 23 and 24. Total porosity tends to be more negatively correlated with m, whereas $T_{2,GM}$ shows a mixed trend—some values are positively correlated with m, while others are negatively correlated. This variability may be attributed to the dominance of either vugs or more connected pores in different samples. Such behavior is consistent with carbonate rocks containing poorly connected vugs.

A similar insight is reflected in Equation 7, which can be reformulated as:

$$m = (\phi_{macro} - 0.1) + 2 \quad (11)$$

In this form, the value 2 represents Archie's standard cementation exponent. When $\phi_{macro} < 0.1$, the cementation factor m is less than Archie's standard, suggesting reduced cementation. Conversely, when $\phi_{macro} > 0.1$, m exceeds the standard value, indicating increased cementation—potentially due to the presence of more vugs or poorly connected porosity.

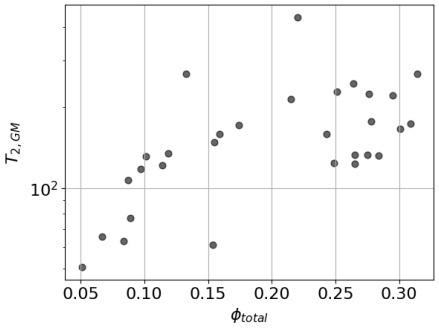


Fig. 22. Correlations between total porosity and $T_{2,GM}$

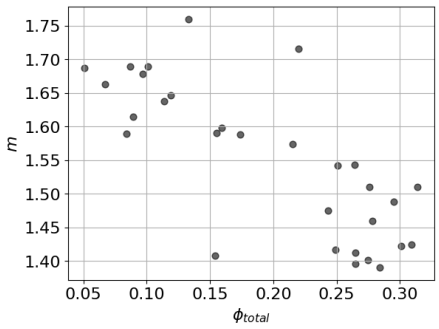


Fig. 23. Correlations between total porosity and m

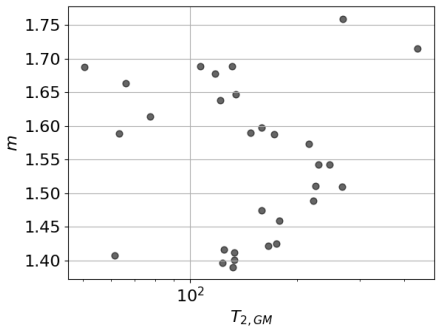


Fig. 24. Correlations between $T_{2,GM}$ and m

Second, the expressions in Equations (7) and (9) involve different tortuosity factors, represented by the numerators in both equations. This difference is not surprising when one acknowledges that tortuosity and

cementation factors are not truly independent, even though they appear to be so in Archie's original empirical formulation. The variation in the *m* distributions in Equations (7) and (9) can thus be seen as two different ways of representing the interplay between cementation and tortuosity factors. Assuming tortuosity to be a fixed constant overlooks the inherent heterogeneity of the rock. Therefore, recognizing that tortuosity and cementation factors are interdependent may offer deeper insights into how rock lithology and porosity influence the formation factor.

6 Conclusion

We present a symbolic regression-based workflow for petrophysical modeling, enabled by the development of a user-friendly software tool. The software integrates multiple open-source SR packages and guides users through data preparation, model generation, and evaluation—making advanced SR techniques accessible to petrophysicists. Applied to both unconventional and carbonate reservoirs, the workflow demonstrates its ability to produce accurate, interpretable models that capture the complex heterogeneities of formations. By bridging the gap between machine learning and practical petrophysical analysis, the software significantly lowers the barrier to adopting symbolic regression in developing reliable and interpretable data-driven petrophysical models.

References

1. W.L. Cava, P. Orzechowski, B. Burlacu, F.O.d. Franca, M. Virgolin, Y. Jin, M. Kommenda, J. H. Moore, Contemporary Symbolic Regression Methods and their Relative Performance, arXiv:2107.14351v1 (2021)
2. D.A. Augusto, H.J.C. Barbosa, Symbolic Regression via Genetic Programming, *6th Brazilian Symposium on Neural Networks, Proceedings*. **Vol. 1.**, 173-178 (2000)
3. S. Chen, W. Shao, H. Sheng, H. Kwak, Use of Symbolic Regression for Developing Petrophysical Interpretation Models, *Petrophysics*, **64 (02)**, 174-190 (2023)
4. S. Chen, C.M. Jones, B. Dai, W. Shao, Developing Live Oil Property Models with Global Fluid Database Using Symbolic Regression, *SPWLA 64th Annual Logging Symposium*, SPWLA-2023-0018 (2023)
5. https://pysr.readthedocs.io
6. A. Kartelj, M. Djukanovic, RILS-ROLS: Robust symbolic regression via iterated local search and ordinary least squares. *Journal of Big Data,* **10(1)** , Article 71 (2023)
7. https://dev.heuristiclab.com/
8. S.M. Udrescu, M. Tegmark, AI Feynman: a Physics-Inspired Method for Symbolic Regression, *Science Advances*, **6(16)** (2020)
9. https://www.datarobot.com
10. G.R. Coates, M. Miller, M. Gillen, G. Henderson, An Investigation of a New Magnetic Resonance Imaging Log, *SPWLA 32th Annual Logging Symposium*, paper DD (1991)
11. G.E. Archie, Electrical Resistivity Log as an Aid in Determining Some Reservoir Characteristics, *Trans. AIME*, **146**, 54-62 (1942)