

## Review of physics-informed machine learning (PIML) methods applications in subsurface engineering

Utkarsh Sinha <sup>a,b</sup>, Birol Dindoruk <sup>b,\*</sup>

<sup>a</sup> Xecta Digital Labs, United States

<sup>b</sup> University of Houston, United States



### ABSTRACT

In recent years, there has been a growing trend in leveraging state-of-the-art machine learning (ML) techniques to solve complex science and engineering problems. While ML serves as a powerful tool to simplify computations, fill in gaps, and handle missing data, it should not be treated as a complete black-box solution. Relying solely on a black-box ML approach poses a significant risk of generating unphysical or inaccurate results. To mitigate these risks and ensure that the model adheres to the governing physical principles, it is essential to integrate machine learning with physics-based constraints, forming what is known as a physics-informed machine learning (PIML) model.

In the oil and gas industry, PIML has emerged as a prominent solution, blending the computational efficiency of ML with the rigor of physics-based methods. Hybrid models, which combine physics-driven frameworks with ML, address key limitations of purely physics-based approaches—such as the lack of sufficient inputs or the high computational expense required to achieve convergence—and the shortcomings of unguided, purely data-driven ML models.

This paper offers a structured overview of application-centric areas in the oil and gas industry where physics-guided ML models and hybrid frameworks have been effectively deployed. It explores various use cases, methodologies for integrating physics into ML, the challenges faced in these implementations, and the advantages and disadvantages of these hybrid approaches compared to traditional physics-based methods.

### 1. Introduction

Physics-based modeling and simulation is a vital part of oil and gas industry where the computer-aided engineering (CAE) tools have been widely used to analyze the physical behavior of artefacts to be created, mainly to validate their requirements. Modeling in the upstream oil and gas industry is the development of a virtual model that mimics/describes the physical characteristics and future performance of subsurface hydrocarbon reservoirs, oil and gas wells, reservoir performance, well performance, economic value of a field, etc. A variety of modeling approaches are used in the oil and gas industry aimed on solving physics based governing equations describing that particular phenomenon. Some of the famous examples of physics based modeling approaches used across the oil and gas industry are, (1) well diffusivity equation (convective transport mass balance (3 phase-3 component, black oil model ([Trangenstein and Bell, 1989a](#); [Trangenstein and Bell, 1989b](#))) equation combined with Darcy's law ([Darcy, 1856](#)) to form a partial differential equations (in space and time)), are combined with PVT properties to solve a black-oil reservoir simulation ([Aziz, 1979](#); [Trangenstein and Bell, 1989a, 1989b](#)) (2)Solving a material balance using tank-model representing a reservoir (generally applicable during

pseudo-steady state) to calculate reservoir pressure as a function of time or to fluid initial fluid in place (3) solving 3-phase multiphase flow solution along the wellbore to generate pressure traverse curves and vertical flow performance curves ([Hagedorn and Brown, 1965](#); [Beggs and Brill, 1973](#); [Mukherjee et al., 1981](#)) (4) Calculation of mud weight and density to keep hydrocarbon in place while drilling (without fracturing the formation) using static pressure calculation ([Okrajni and Azar, 1986](#)). (5) Non-Newtonian viscosity model ([Herschel and Bulkley, 1926](#)) combined with energy equation laminar/turbulent steady state flow models to find pressure drop loss in the annulus, to calculate the equivalent circulation density ([Okrajni and Azar, 1986](#)). In all these examples as well many more the underlying physics-based equations are solved analytically/numerically (depending on the complexity) either through using simulator/calculator code written in languages like MATLAB, FORTRAN, C#, etc. or solved on excel spreadsheet or solved through a commercial software where the simulator/calculator code is wrapped around with graphic user interface (GUI) after several rounds of testing and validation of the code on known cases.

One of the known disadvantages of physics-based modeling is the complexity of solving underlying governing equation, which may require to make several assumptions for simplification, that leads

\* Corresponding author.

E-mail address: [bdindoru@central.uh.edu](mailto:bdindoru@central.uh.edu) (B. Dindoruk).

inherent biases and therefore could lead to high errors when these assumptions aren't fully valid, on top of that, numerical solutions could be very computationally expensive and take several hours (and sometimes even a couple of days) before reaching convergence, when results are generated. For example, in compositional reservoir simulation if you are attempting to solve PDE in space and time (Convection-Dispersion equation (Orr, 2007)) using finite difference method for each component. Another disadvantage of use a full-blown physics model is its demand on high amount on information as inputs, to be initialized, some of these inputs could be difficult obtain and may bear measurement errors/uncertainties if measured under unfavorable conditions, which can offset the physics-based models. Due to these indicated disadvantages of pure physics-based models, there arises need of empirical and semi-empirical/semi-analytical methods (aka hybrid models) (Zendehboudi et al., 2018; Rai et al., 2020; Mosavi et al., 2019) which could generate accurate results in short time span, with limited inputs in hand without disobeying/exceeding physics-based constraints on the outputs.

One of the key methodologies of empirical predictive modeling is the application of machine learning models (ML) to establish a nonlinear relationship between a set of inputs and outputs, if correlatable (See Fig. 1). Machine learning models "Contrary to the traditional approach, where first principles models are used, machine learning fits empirical models using experimental data (training data). A proper data split is necessary to introduce the right amount of model complexity and avoid overfitting" (Mowbray et al., 2022). The framework for developing and evaluating a robust machine learning model, as presented in Alsabaa et al. (2022), is summarized in Fig. 2. Petroleum engineering is perhaps one of the most interdisciplinary engineering fields which deals with complex and many times with problems with incomplete information. Upstream Oil and Gas is embracing Digital Transformation for the purpose of improving operational performance in every front. From reservoir to market, oil and gas industry deal with various types of data from volume of data to scale dependency of the data, therefore machine learning models have been widely used in the oil and gas industry in recent years, both academically and industrially (Balaji et al., 2018). If we use the keyword "Machine Learning Model" onepetro (See Fig. 3) and look for peer-reviewed/conference proceedings articles, the oldest article dates to 1973 (Haenel, 1973). Between 2010 and 2022, 1286 articles show up under the search keyword "Machine Learning Model", on onepetro. Some of the famous supervised machine learning methods frequently used in the oil and gas industry based on the number of total papers present on onepetro, have been shown in Fig. 3. Machine Learning models are prevalent in the field of Reservoir Engineering (Mohaghegh, 2017; Nwachukwu et al., 2018; Pachaliev et al., 2022), Reservoir Fluid Phase Behavior (Oloso et al., 2017), Petrophysics and Reservoir Characterization (Anifowose et al., 2015, 2017), Production Engineering (Ahmadi and Chen, 2019; Wang and Chen, 2019) and Drilling Engineering (Hegde et al., 2015; Noshi et al., 2018).

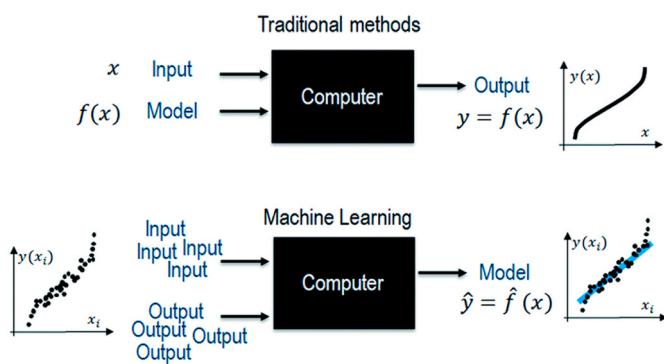


Fig. 1. Workflow of a machine learning regression model (Mowbray et al., 2022).

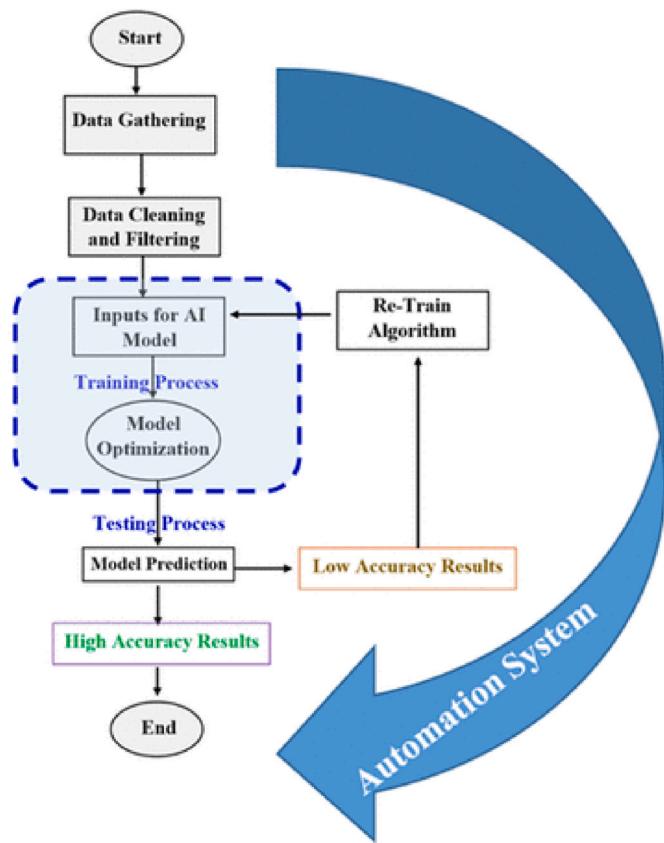


Fig. 2. Framework for creating and testing a robust machine learning model (Alsabaa et al., 2022).

One of the biggest disadvantages of using machine learning models in gray-box/back-box manner over pure physics-based models is the amount of freedom that the machine learning models have when they predict on an unseen dataset, they are not constrained or guided in any way and therefore they often end up predicting unphysical results. This issue is quite well known but another lesser noticed disadvantage of machine learning models is that they could predict physical results for unphysical inputs. Physics-based models are designed based on the physics-based governing equations and therefore it will not be able with unphysical inputs, for example the material balance results will turn become unphysical if you assume a negative solution gas-oil ratio, formation volume factor, etc., however, that might not be the case if you use an unconstrained and unguided machine learning model to calculate the results from material balance (oil in place vs time or average reservoir pressure vs time). A comparison of pure physics-based models and naïve machine learning models for production and reservoir engineering applications is shown in Fig. 4, based on advantages and disadvantages of each of the methods.

To solve the problems stated above, it becomes very important to impose physics-based constraints on the machine learning models or guide the machine learning models in a way that they don't disobey the underlying physics and don't produce unphysical results. Such models where physics is blended to constraint/guide a naïve machine learning model to make accurate and physical models (these models honor the underlying physics), which produce accurate results even on unseen data on testing, are referred to in different names like physics-informed models (PIML), hybrid models, data-driven reduced order models, physics-guided machine learning models, etc. On combining the principles and key advantages physics and machine learning we take benefits from best of the two worlds, resulting in superior ability to work on limited number of inputs and relatively small dataset size (NDPTS) (see Fig. 5), computationally fast once trained, highly scalable, ability to

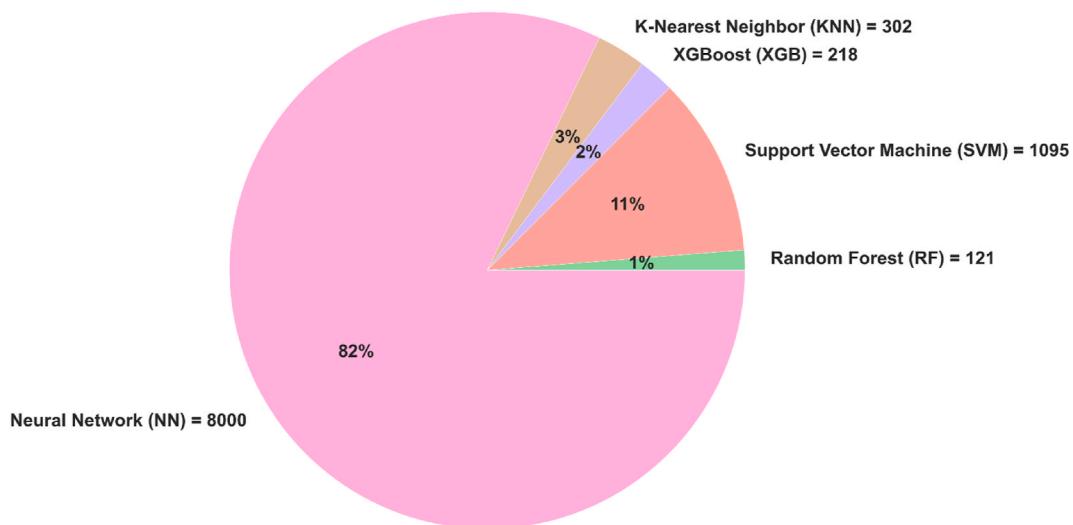


Fig. 3. Frequently used supervised ML methods used in the papers published in onepetro.

## Physics-based Model

- Pros**

  - Requires less calibration data
  - Reliable over the lifetime of an asset
  - Can be used to model other operational problems
  - Can be used to estimate unmeasured variables

- Cons**

  - Requires deep domain knowledge
  - High computational cost
  - Dependent on PVT data accuracy

## Machine-Learning Model

- Pros**

  - Does not require deep knowledge about physics
  - Can make use of all available sensor data
  - Low computational cost once trained
  - Easy to deploy for many wells

- Cons**

  - Not suitable with limited historic data
  - Limited operational experience
  - Re-training required when operational conditions change

Fig. 4. Pros and cons of physics-based and ML-Based methods for production and reservoir engineering applications ([Zongchang Yang, 2020](#))

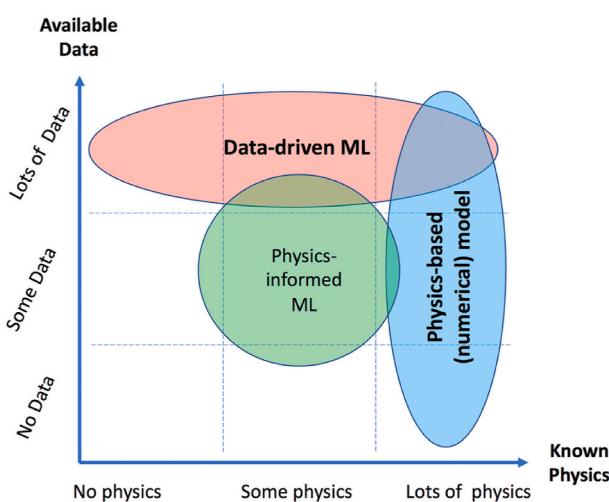


Fig. 5. Benefits of physics-informed machine learning models ([Tartakovsky et al., 2020](#)).

capture non-linear and complex relationships between inputs and outputs, generate extra bit of information from the dataset like inferring physical relations or discovering the governing equation, and lastly as

we discussed before they honor the underlying physics.

In the next few sections, we discuss some of the key approaches to combine physics with machine learning, major application in the field of in the field of petroleum and some of the case studies where physics-informed machine learning models have been successfully deployed to solve complex petroleum engineering problems.

## 2. Methodologies to combine ML and physics

### 2.1. Physics-based dataset screening and cleansing

Physics-based dataset screening is a crucial method for enhancing the quality of data used to train machine learning (ML) or deep learning (DL) models. A model's performance is fundamentally dependent on the quality of the data it is trained on. Traditional screening techniques in ML or DL typically involve excluding data points with missing values and eliminating obvious outliers—values that deviate significantly from the expected distribution of the dataset. However, physics-based data screening goes beyond these conventional methods by integrating domain expertise and a deep understanding of the underlying physical processes being modeled.

This approach leverages knowledge of physical phenomena, which enables practitioners to define valid ranges of parameters based on what is physically possible. It also accounts for the inherent limitations and uncertainties tied to the experimental conditions under which the

dataset was generated. For instance, specific physical properties, constraints, or theoretical bounds are used to guide data cleansing and ensure the dataset aligns with real-world scenarios.

An example of physics-based data screening was demonstrated in [Sinha et al. \(2021\)](#), where a super learner model for predicting minimum miscibility pressure (MMP) was developed. In this case, the dataset was refined in two important ways: (1) Only MMP data points derived from slimtube experiments were included, as these experiments are the most consistent with the true definition of MMP. (2) Data points were further filtered to exclude any instance where the measured MMP was less than the bubble point pressure ( $P_{bubble}$ ), a condition that contradicts the underlying physics of miscibility.

By incorporating domain knowledge, physics-based screening ensures that only high-quality, relevant, and physically valid data points are used for model training, leading to more robust and accurate ML/DL models.

## 2.2. Physics-based feature engineering

The selection of inputs or features is a critical aspect of training any machine learning (ML) or deep learning (DL) model, including deep neural networks (DNNs). In the context of modeling physical phenomena, choosing the right features—those that are well-understood from physics and are most relevant to the problem at hand—is essential for building a successful model. Often, these features are not readily available in the dataset and must be generated using physical equations or derived relationships.

Physics-based feature engineering involves creating new inputs that capture key aspects of the system under study, often through domain-specific equations or models. For example, in [Sinha et al. \(2021\)](#), a physics-based feature was engineered by calculating vapor pressure from reservoir temperature to model minimum miscibility pressure (MMP). This process of feature creation ensures that the model incorporates critical physical relationships that drive the underlying phenomena, making it more accurate and reliable.

In the oil and gas industry, the importance of physics-based feature engineering has been well documented in the literature. Studies such as [Sankaran et al. \(2020\)](#), [Staff et al. \(2020\)](#); [Molinari and Sankaran \(2021a,b\)](#) have explored how domain knowledge can guide the development of features that enhance the predictive power of ML/DL models. A notable example of this can be seen in the recent work by [Moncayo-Riascos et al. \(2022\)](#), where physics-based feature selection was employed to model asphaltene onset pressure (AOP). [Fig. 6](#) in that study illustrates how careful selection of inputs grounded in physical theory can significantly improve model performance.

By incorporating physics-based knowledge into the feature selection

and engineering process, ML and DL models can more effectively capture the complexities of physical systems, resulting in better predictions and insights.

Physics-based transformations of features or outputs play a critical role in enhancing the accuracy and physical relevance of machine learning (ML) and deep learning (DL) models. These transformations leverage known physical laws, empirical observations, or experimental data to refine how input features or outputs are represented, allowing models to capture the underlying physics more effectively.

An example of such transformations is shown in Equations (1) and (2):

$$g(Y) = f(X) \quad (\text{Eq.1})$$

where  $X$  represents the input matrix,  $f(x)$  is the fitted model on  $X$ ,  $Y$  is output and  $g(Y)$  is a transformation function on  $Y$ .

$$Y = f(X_1, X_2, g(X_3)) \quad (\text{Eq.2})$$

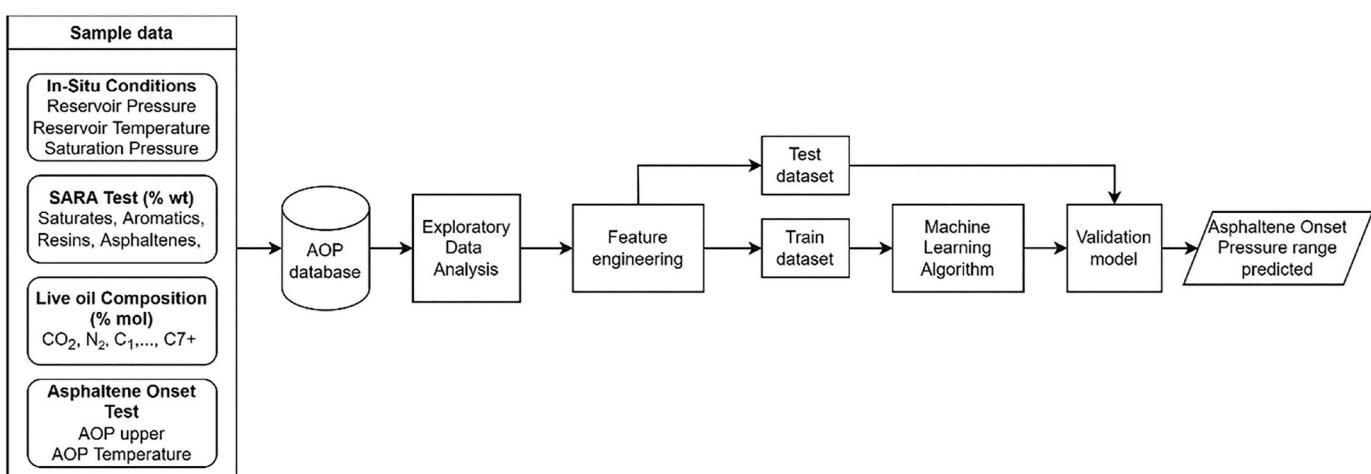
where  $X_1, X_2, X_3$  are input arrays,  $f(x)$  is the fitted model based on inputs,  $Y$  is output and  $g(X_3)$  is transformation function applied specifically to the third input array ( $X_3$ ).

These transformations are often inspired by physical principles, well-established equations, or known trends observed through experiments. For instance, in [Sinha et al. \(2022\)](#), a double logarithmic transformation was applied to the output variable of viscosity. This transformation was motivated by the work of [Ali, 2003](#), which demonstrated a linear relationship between the logarithm of viscosity and reservoir temperature based on experimental data. The double logarithmic transformation allowed the model to more accurately capture this physical trend, ultimately improving model performance.

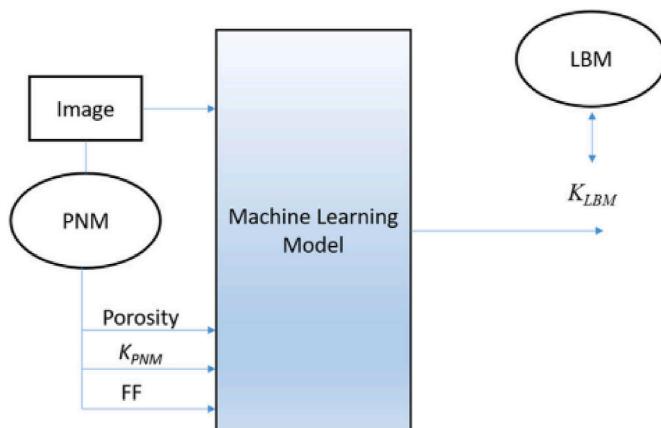
In addition to transforming outputs, the use of simulated data derived from physics-based models is gaining traction in the oil and gas industry as part of physics-informed feature engineering. This approach allows for the generation of high-quality datasets that respect physical constraints, filling gaps in experimental data and improving the robustness of ML models.

One notable example of this practice can be seen in [Tembely et al. \(2020\)](#), where a pore network model (PNM) was used to simulate certain input parameters that were then fed into an ML model for predicting rock permeability (see [Fig. 7](#)). Similarly, [Zhao et al., 2020](#) utilized a PNM model to generate missing values in an input dataset obtained from special core analysis (SCAL) measurements. The enhanced dataset, enriched by physics-based simulated values, was subsequently used to predict relative permeability via an ML model (see [Fig. 8](#)).

These examples illustrate how physics-based transformations and simulated datasets can significantly improve the quality and relevance



**Fig. 6.** Physics-based feature engineering for AOP prediction using ML model ([Moncayo-Riascos et al., 2022](#)).



**Fig. 7.** ML model features generated from physics model to predict absolute permeability (Tembely et al., 2020).

of ML model inputs and outputs. By grounding model training in physical reality, these methods ensure that the resulting predictions are not only statistically sound but also physically consistent with the underlying processes they aim to represent.

### 2.3. Physics-based continuity imposition

The practice of ensuring that two separate models trained to describe different regions of a physical phenomenon transition smoothly at a defined switch point. This switch point is determined based on prior physical knowledge, allowing the models to handle distinct phases or regimes of a system. A common example is modeling live oil viscosity for under-saturated and saturated oil, with the switch occurring at the bubble point pressure (Dindoruk and Christman, 2004).

In such cases, it's important to not only create accurate models for each regime but also impose continuity—and sometimes differentiability—between the models at the switch point. This ensures a smooth transition between the two regimes, preserving the physical realism of the overall model. Without this continuity, the model might exhibit unrealistic jumps or discontinuities at the transition point,

leading to erroneous predictions and reduced reliability.

Despite the importance of continuity imposition, this approach has not gained much attention in machine learning (ML) literature. There is little to no significant work in ML that explicitly been used in physics-based correlations. For instance, Yuan et al. (2004) employed this approach when developing a minimum miscibility pressure (MMP) correlation, ensuring smooth transitions at a methane ( $\text{CH}_4$ ) impurity level of zero. Similarly, Sinha et al., 2022 applied this concept to ensure continuity at viscosity switch points between light oils (See Fig. 9). While these examples demonstrate the importance of continuity imposition, they are both from empirical correlations, not ML models (see Fig. 9a and b).

This approach could be extended to ML-based models, especially in cases where distinct physical regimes need to be modeled separately but must transition smoothly, preserving the physical integrity of the phenomenon across different input conditions.

### 2.4. ML inverse modeling

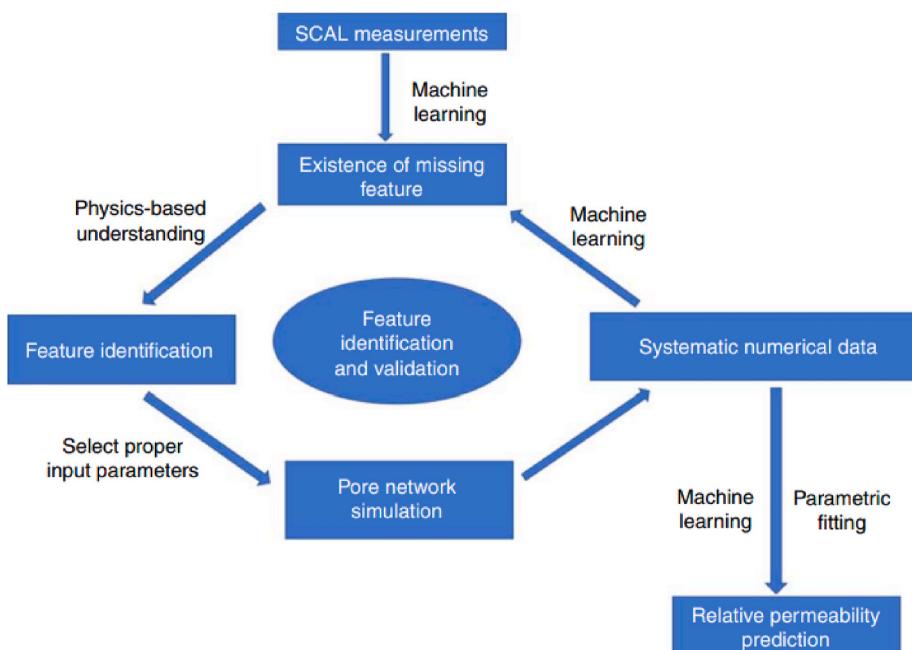
ML inverse modeling is a process where the output of a system—either measured or generated through a physics-based model—is used to reverse-engineer the key input parameters that describe the system. This approach contrasts with traditional forward modeling, where inputs are used to predict outputs. In inverse modeling, we already know the final outcome and use it to uncover the underlying factors or causes (i.e., the input parameters) that led to the observed results. The goal is to develop an inverse model that maps from the outputs back to the inputs, thereby enhancing our understanding of the system.

Mathematically, inverse modeling can be expressed as:

$$X = f^{-1}(Y) \quad (\text{Eq.3})$$

where  $X$  represents the input parameters to be determined,  $Y$  is the known results dataset (either measured in the field or generated by a physics model) and  $f^{-1}(Y)$  is the inverse ML model that takes  $Y$  as input and calculates  $X$ . The inverse model  $f^{-1}(Y)$  is trained on known pairs of inputs  $X$  and outputs  $Y$ , allowing it to learn the mapping from outputs to inputs.

An example of inverse modeling in reservoir engineering might involve the following.



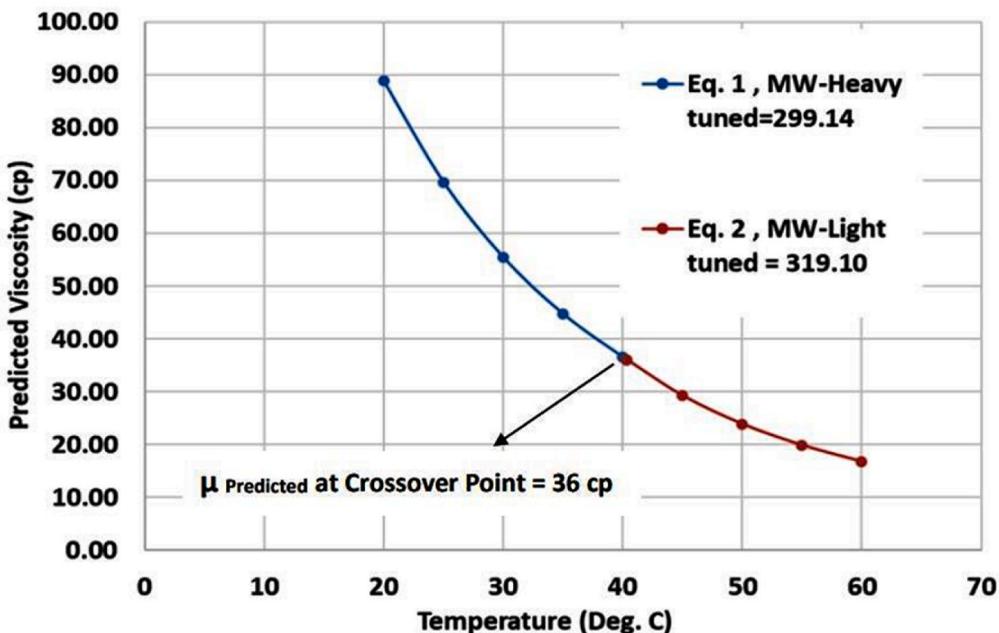
**Fig. 8.** ML model features generated from physics model to predict relative permeability (Zhao et al., 2020).

**Explicit Method Continuity at Crossover Point.** Table 6 illustrates the algorithmic use of the Explicit Method (Eqs. 1 and 2) so that the continuity is retained for a given oil at the crossover point (if present) between the two branches over a range of temperatures.

<b>Case 1</b>	Step 1. Tune the MW (obtain MW-Heavy <sub>tuned</sub> ) to match the reference viscosity ( $\mu_{Ref}$ ) with the viscosity predicted by Eq. 1. Step 2. Use the MW-Heavy <sub>tuned</sub> to predict viscosities at other temperatures * No Crossover point here.*
<b>Case 2</b>	A) When Reference Viscosity ( $\mu_{Ref}$ ) >36 cp Step 1. Tune the MW (obtain MW-Heavy <sub>tuned</sub> ) to match the reference viscosity ( $\mu_{Ref}$ ) with the viscosity predicted by Eq. 1. Step 2. Using the MW-Heavy <sub>tuned</sub> in Eq. 1 find the temperature [ $T_{Switch}$ (°C)] at which the viscosity predicted by Eq. 1 is equal to 36 cp. Step 3. Tune the MW (obtain MW-Light <sub>tuned</sub> ) to match the viscosity predicted by Eq. 2 at $T_{Switch}$ to be 36 cp. Step 4. For all temperatures below $T_{Switch}$ , Eq. 1 should be used with MW-Heavy <sub>tuned</sub> to predict the viscosity. Step 5. For all temperatures above $T_{Switch}$ , Eq. 2 should be used with MW-Heavy <sub>tuned</sub> to predict the viscosity.  B) When Reference Viscosity ( $\mu_{Ref}$ ) ≤36 cp Step 1. Tune the MW (obtain MW-Light <sub>tuned</sub> ) to match the reference viscosity ( $\mu_{Ref}$ ) with the viscosity predicted by Eq. 2. Step 2. Using the MW-Light <sub>tuned</sub> in Eq. 2 find the temperature [ $T_{Switch}$ (°C)] at which the viscosity predicted by Eq. 2 is equal to 36 cp. Step 3. Tune the MW (obtain MW-Heavy <sub>tuned</sub> ) to match the viscosity predicted by Eq. 1 at $T_{Switch}$ to be 36 cp. Step 4. For all temperatures below $T_{Switch}$ , Eq. 2 should be used with MW-Heavy <sub>tuned</sub> to predict the viscosity. Step 5. For all temperatures above $T_{Switch}$ , Eq. 1 should be used with MW-Light <sub>tuned</sub> to predict the viscosity.

Table 6—Algorithm to be followed for the application of Explicit Method (Eqs. 1 and 2).

(a) – Workflow



(b) – Case Study/Example

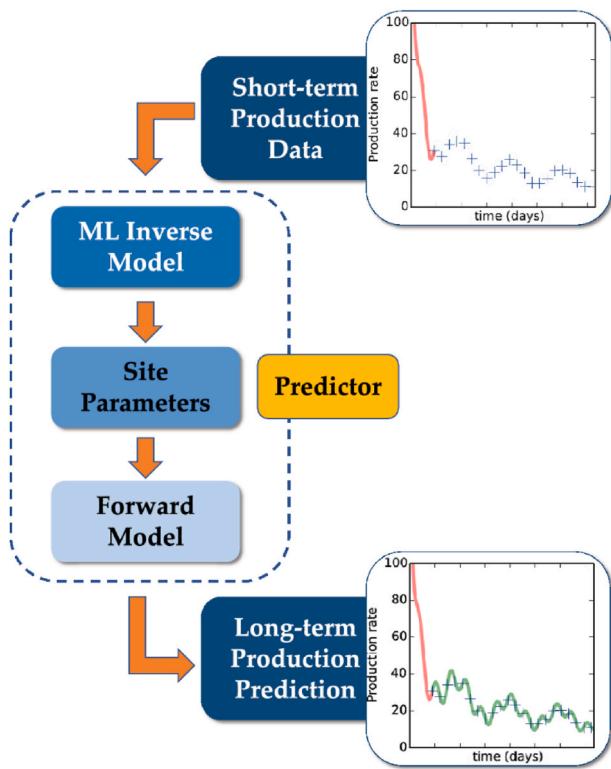
Fig. 9. (a & b): Workflow and example of physics-based continuity imposition between two equations from [Sinha et al. \(2022\)](#).

- X = [Average Reservoir Pressure or  $P_{avg}^{Reservoir}(t)$ , Productivity Index or  $PI(t)$ , Bottom Hole Pressure or  $Pwf(t)$ ], representing the input parameters.
- Y = [Oil, Gas and Water production rates or  $q(t)$ ], representing the outputs.

In this scenario, the inverse model is trained to predict key reservoir parameters such as pressure and productivity index based on the observed production rates. This technique is especially valuable in scenarios where direct measurement of reservoir parameters is challenging or impossible, and the only available data comes from production outputs. The key benefit of ML inverse modeling in this scenario is to infer critical reservoir parameters without the need for costly or time-consuming direct measurements. This allows for faster decision-making, as production data is often readily available in real-time, enabling continuous updates to reservoir models. Moreover, it helps optimize field operations by improving the accuracy of

forecasts and reducing uncertainties in reservoir management, ultimately leading to more efficient resource extraction and enhanced production performance. Similar work can be seen in [Zalavadia et al. \(2022\)](#).

A practical application of ML inverse modeling was demonstrated by [Srinivasan et al. \(2021\)](#) (shown in Fig. 10). They employed field data to generate short-term production forecasts using a high-fidelity physics model, specifically a Discrete Fracture Network (DFN) model. The short-term production data, in turn, served as input for their ML inverse model, which back-calculated essential reservoir parameters required for a reduced-order forward model. This reduced-order model, developed by [Patzek et al. \(2013\)](#), was then used to forecast long-term production for an unconventional reservoir. The integration of inverse modeling in this workflow allowed the authors to generate long-term forecasts with greater efficiency, overcoming the computational complexity of high-fidelity simulations by leveraging the ML model to infer critical inputs.



**Fig. 10.** ML inverse modeling used for long term production forecasting (Srinivasan et al., 2021).

This approach is particularly useful in complex reservoir settings, such as fractured or unconventional reservoirs, where direct forward modeling may be computationally expensive or require detailed input data that is difficult to measure in the field. By using ML inverse models to infer these parameters from readily available production data, it becomes possible to make faster and more accurate forecasts.

## 2.5. Reduced order machine-learning modeling

Dimensionality reduction has gained significant attention for visualizing datasets in low-dimensional space, simplifying the observation of input-output relationships, and facilitating surface fitting (e.g., 3D surface, 2D line, or nonlinear hyperplane). One widely used technique is Principal Component Analysis (PCA) (Wold et al., 1987; Ringnér, 2008), which performs truncated eigenvalue decomposition on the covariance matrix of a dataset. In PCA, eigenvectors represent the principal components, and eigenvalues correspond to the variance captured by each component. Sorting the eigenvalues in descending order and retaining those above a threshold reduces dimensionality. PCA's popularity stems from its objectivity and its effectiveness in capturing dominant variance in the data.

For dynamic systems (e.g., transient fluid flow), where variables like velocity and pressure evolve over time and space (e.g.,  $(u(\vec{x}, \vec{y}, \vec{z}, t)$  or  $P(\vec{x}, \vec{y}, \vec{z}, t)$ ), Reduced Order Modeling (ROM) accelerates flow predictions by generating low-dimensional surrogate models for complex, nonlinear, multi-scale systems. ROM techniques such as Proper Orthogonal Decomposition (POD) (Chatterjee, 2000; Liang et al., 2002; Müller, 2008; POD and DMD, 2022) and Dynamic Mode Decomposition (DMD) (Schmid, 2010; Tu, 2013; Brunton and Kutz, 2022) have proven particularly effective for such dynamic systems.

POD, conceptually similar to PCA, decomposes a spatial-temporal field  $v(x,t)v(\mathbf{x}, t)v(x,t)$  into orthogonal spatial modes and time-dependent coefficients by performing eigenvalue decomposition on the covariance matrix of snapshot data. The field is expressed as a

weighted sum of spatial modes spatial modes  $\phi_k(x)$ , modulated by temporal coefficients  $a_k(t)$ , capturing the system's dominant energy components:

$$u(x, t) = \sum_{k=1}^m a_k(t) \phi_k(x) \quad (\text{Eq.4})$$

where  $\phi_k(x)$  is the matrix of all the eigen vectors (mutually orthonormal) arranged as columns, ordered by descending eigen value magnitudes, also referred as spatial nodes (similar to principal components).

"We refer to modal decomposition as a mathematical technique to extract energetically and dynamically important features of fluid flows. The spatial features of the flow are called (spatial) modes and they are accompanied by characteristic values, representing either the energy content levels or growth rates and frequencies. These modes can be determined from the flow field data or from the governing equations. We will refer to modal decomposition techniques that take flow field data as input to the analysis as data-based techniques." (Taira et al., 2017).

POD has notable limitations. Its emphasis on capturing the most energetic features can lead to the neglect of dynamically significant but lower-energy modes, which can be crucial in some systems. Additionally, POD requires access to the entire dataset, or "snapshots," collected over time before performing the decomposition. This precludes its use in real-time applications where data is continuously generated, as POD cannot adapt to new data dynamically. Moreover, the static nature of the POD basis may struggle to accurately represent evolving dynamics in highly nonlinear systems.

While POD provides a time-dependent representation of the system through its modes, it does not predict future states directly. POD extracts modes that describe how the system has behaved historically but does not inherently capture the dynamic evolution of these modes over time. To predict future states, you need to understand how these modes change in response to system dynamics. This is because POD gives you a representation of the system at a given time but lacks the mechanism to project forward into the future based on the evolving dynamics.

However, POD has limitations. Its primary focus on maximizing energy content can sometimes overlook dynamically significant modes that have lower energy but are crucial for understanding system behavior. This limitation is particularly relevant in systems where these low-energy features play an essential role. Additionally, POD requires access to the entire dataset, or "snapshots," collected over time before performing the decomposition (Chatterjee, 2000). This precondition makes POD less practical for real-time applications where data is continuously generated or updated, as it does not adapt to new data.

Moreover, the static nature of the POD basis can struggle to capture evolving dynamics in highly nonlinear systems. Although POD provides a time-dependent representation of the system through its modes, it does not inherently account for how these modes evolve over time. To predict future states of the system, one needs to understand the dynamics governing the changes in these modes. POD does not provide this dynamic information directly but rather offers a snapshot of the system's behavior based on historical data.

To address these issues, Galerkin projection is often employed. This technique projects the governing partial differential equations (PDEs) onto the reduced POD basis. By doing so, it transforms the high-dimensional system into a lower-dimensional set of ordinary differential equations (ODEs), significantly reducing computational costs while retaining essential dynamics (Berkooz et al., 1993; Rapun and Vega, 2010; Akhtar et al., 2009; Ullmann, 2014).

Galerkin-projected POD has been effectively used in fields such as fluid dynamics, structural analysis, and heat transfer, particularly in scenarios requiring repeated simulations, like parametric studies, optimization, and control (Lumley, 1970; Rowley et al., 2004). The use of this technique for reservoir simulation and optimization was demonstrated by Voloskov and Pissarenko (2021), who employed an adaptive POD-Galerkin method to enhance the efficiency and accuracy of

reservoir models. By focusing on the most significant modes, this method offers a computationally efficient way to approximate complex systems with minimal accuracy loss, making it suitable for real-time predictions, control, optimization, and uncertainty quantification.

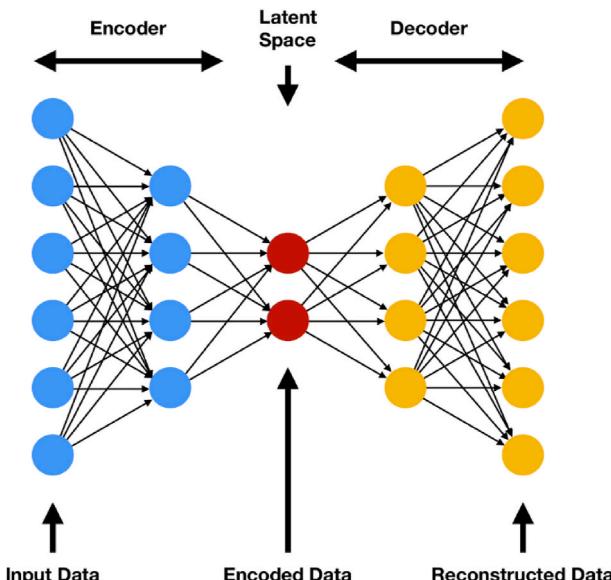
In contrast to POD, Dynamic Mode Decomposition (DMD) is better suited for time-dependent systems. DMD extracts modes from the eigenvalue spectrum associated with a system's underlying dynamics, revealing growth rates, oscillations, and frequencies directly from the data. This makes DMD highly effective for analyzing transient behaviors. DMD can approximate nonlinear systems by constructing a linear map in a higher-dimensional space, and since it operates directly on time-series data without requiring a complete dataset upfront, it is well-suited for real-time applications. DMD also provides insights into the temporal evolution of modes, offering a dynamic understanding of system behavior compared to the static POD basis, making it the preferred technique for many real-world applications involving complex, time-varying systems. For example, in Zalavadia et al. (2019), where it was used to predict pressure and saturation state solutions for new bottom-hole pressure profiles, optimizing production control. The methodology effectively selects an optimal reduced set of dimensions to ensure stability, improving both speed and predictive accuracy for real-time reservoir management.

Recent advancements in fluid mechanics have leveraged machine learning (ML) and deep learning (DL) techniques to develop efficient reduced-order models (ROMs). These methods are particularly valuable as they enable the creation of highly effective and accurate ROMs without requiring detailed high-fidelity model equations or process information. When available, incorporating process knowledge can guide ML or DL models, enhancing their performance through physics-informed approaches.

Autoencoders (Cho et al., 2014; TowardsDS, 2022) are a prominent DL technique used for reduced-order modeling. These deep neural networks (DNNs) are trained on high-dimensional datasets, which could originate from extensive input matrices or high-fidelity simulations of dynamic systems. An autoencoder consists of three main components: the encoder, the code layer, and the decoder.

The Encoder (depicted in blue in Fig. 11) consists of multiple interconnected layers of neurons. Its role is to transform the high-dimensional input data into a lower-dimensional representation, known as the latent space.

The Code Layer (shown in red in Fig. 11) serves as a bridge between



**Fig. 11.** Deep neural network (DNN) architecture of an autoencoder (Softcrylic et al., 2022).

the encoder and decoder. This single layer contains a reduced number of nodes that represent the compressed version of the dataset.

The Decoder (illustrated in yellow in Fig. 11) is generally symmetric to the encoder and reconstructs the original high-dimensional inputs from the reduced-dimensional latent space. The decoder layers mirror the encoder layers to facilitate accurate reconstruction (refer to Fig. 12).

The loss function used in autoencoders aims to minimize the reconstruction error and can be expressed as:

$$\text{Loss} = \text{Loss}^{\text{Reconstruction Accuracy}}(X^{\text{Actual}}, X^{\text{Reconstructed}}) + \alpha_1 \text{Loss}^{\text{Complexity}} \quad (\text{Eq.5})$$

- $X^{\text{Actual}}$  represents the original input data, and  $X^{\text{Reconstructed}}$  is the data reconstructed by the decoder.
- $\text{Loss}^{\text{Reconstruction Accuracy}}$  quantifies the reconstruction error, which can be measured using metrics such as Mean Squared Error (MSE) or Absolute Relative Error (ARE).
- $\text{Loss}^{\text{Complexity}}$  accounts for regularization to prevent overfitting, with  $\alpha_1$  as its weight.

If physical constraints are applied, the loss function can be extended as:

$$\text{Loss} = \text{Loss}^{\text{Reconstruction Accuracy}}(X^{\text{Actual}}, X^{\text{Reconstructed}}) + \alpha_1 \text{Loss}^{\text{Complexity}} + \alpha_2 \text{Loss}^{\text{Physics}} \quad (\text{Eq.6})$$

In this modified function.

- $\text{Loss}^{\text{Physics}}$  incorporates constraints based on physical principles, with  $\alpha_2$  as its weight.

Each of the three components of the Loss function is multiplied with a weight based on their relative importance ( $\alpha_1$  and  $\alpha_2$ ). By imposing physics based constrain our reduced order modeling approach becomes even more physics-informed.

Autoencoders can effectively reduce the dimensionality of both discrete datasets and time series data. For discrete datasets, dense layers are typically used, whereas time series data may require layers designed for temporal sequences, such as Recurrent Neural Networks (RNNs) (Sherstinsky, 2020) and Long Short-Term Memory (LSTM) networks (Sherstinsky, 2020).

Dimensionality reduction via autoencoders simplifies visualization and pattern discovery in high-dimensional systems. The reduced-dimensional latent space enables efficient application of regression models, which have lower computational complexity compared to models operating on full-dimensional data:

Regression with full dimensions:

$$Y = f(X) \quad (\text{Eq.7})$$

Where  $X$  is  $m \times n$  matrix of inputs.

Regression with latent space:

$$Y = f(X_r) \quad (\text{Eq.8})$$

Where  $X_r$  is  $m \times n_r$  matrix where  $n_r < n$

Reducing dimensionality thus simplifies the regression process, as demonstrated by Liu et al. (2015), who used latent space features for classification tasks related to well failure prediction. Similarly, latent state features from previous time steps can be used for time series prediction, allowing for simpler models that forecast current states from historical data (Kaggle1, 2022).

The reconstruction loss can also be employed for anomaly detection. By analyzing the reconstruction errors, one can identify deviations from expected behaviors in dynamic systems. For example, as shown in Fig. 13, Nakagawa et al. (2021) utilized long short-term memory layer autoencoder (LSTM-AE) for reconstructing an input dataset matrix consisting of 13 key parameters of drilling (Top drive rotation speed,

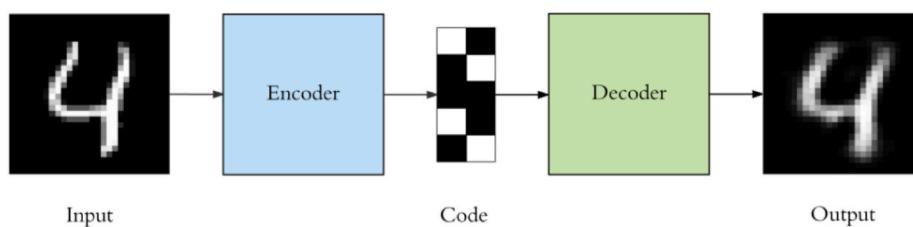


Fig. 12. Workflow of an autoencoder (Medium, 2017)

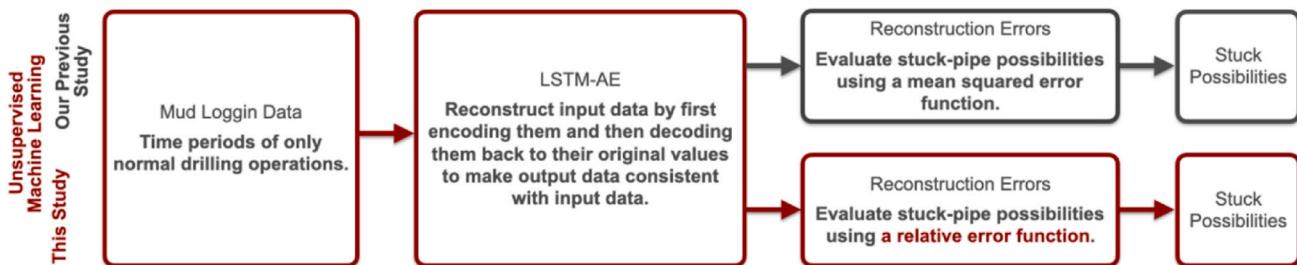


Fig. 13. Autoencoder used for detecting stuck pipe anomalies (Nakagawa et al., 2021).

Average rate of penetration, Top drive torque, etc.) and based on the reconstruction error they were able to detect a possible stuck pipe anomaly (See Fig. 13).

In summary, autoencoders offer a robust framework for dimensionality reduction, anomaly detection, and regression modeling, making them a powerful tool in dynamic systems modeling.

In addition to traditional autoencoders, Variational autoencoders (VAEs) represent an advanced form of autoencoders designed to model complex data distributions. Unlike traditional autoencoders, which encode inputs into fixed representations, VAEs encode data as probabilistic distributions. This approach is achieved through a regularization process that prevents overfitting and allows the latent space to capture essential properties of the data distribution, such as mean and variance in the case of Gaussian distributions. Consequently, VAEs can generate new data samples that adhere to the underlying distribution of the original dataset.

The generative capabilities of VAEs are particularly advantageous for several applications. In reservoir modeling and simulation, for example, VAEs can capture subsurface heterogeneity more effectively, leading to more accurate simulations and improved management strategies. This capability is crucial for optimizing exploration and production by analyzing geological data to identify promising drilling locations. Additionally, VAEs enhance predictive maintenance by forecasting equipment failures from sensor data, thus improving asset reliability. In geology, machine learning models, including VAEs, assist in mapping complex sedimentary facies and addressing challenges related to non-stationary and location-specific facies.

Moreover, VAEs extend the functionality of traditional autoencoders by incorporating probabilistic elements into the encoding process. This enhancement allows VAEs to synthesize new data samples that are similar to the training data, making them valuable for tasks involving data augmentation and synthesis.

For a more detailed understanding of VAEs and their applications in generative modeling, see the foundational works by Kingma (2013) and Rezende et al. (2014), which provide comprehensive insights into the theory and implementation of VAEs. Additional references include McCoy et al. (2018), Śmieja et al. (2020), and Qiu et al. (2020), which explore applications of VAEs in various fields such as physics modeling and geology.

## 2.6. Simulated results combined ML model

### 2.6.1. Proxy/surrogate model using simulated results

Proxy or surrogate serve as computationally efficient alternatives to running full high-fidelity simulations using physics-based models, which can be computationally expensive. In this approach, a physics model is used to generate results for a broad range of physically plausible input combinations. These results are then used to train a machine learning (ML) or deep learning (DL) model to establish a correlation between the inputs and the simulated outcomes.

Traditional Proxy Modeling (TPM) and Smart Proxy Modeling (SPM) workflows using simulated results are summarized in Fig. 14, based on Bahrami et al., 2022. In TPM, the focus is on creating a straightforward surrogate model that approximates the physics-based simulations. SPM, on the other hand, incorporates advanced feature engineering techniques to generate new parameters that capture the complexities within the reservoir, including constraints and grid characteristics. This approach enables the model to better handle the nuances of complex reservoir simulations.

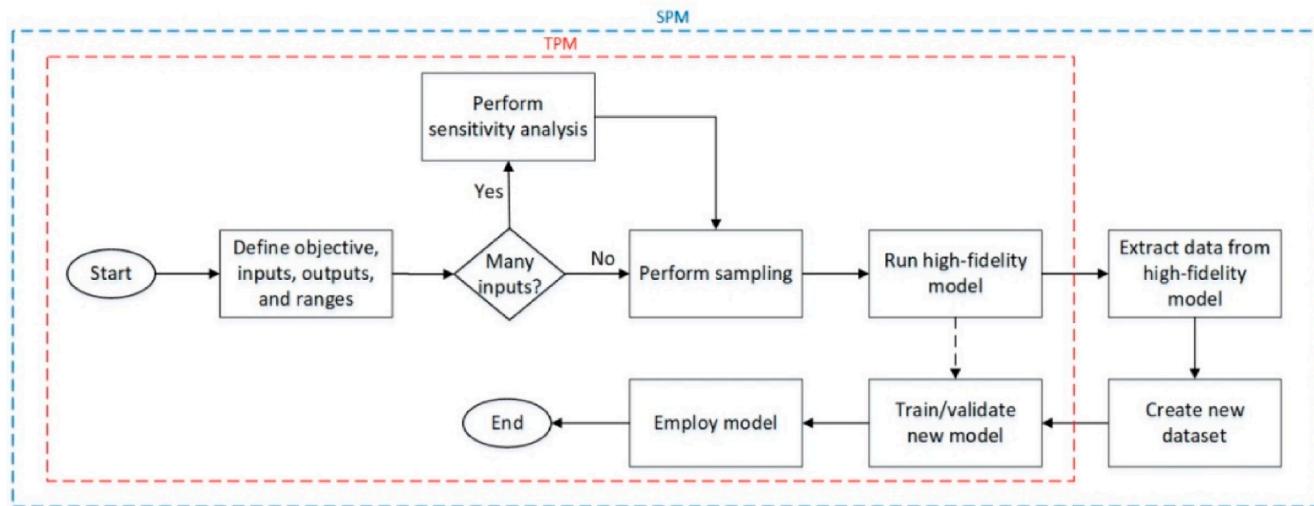
An example of proxy modeling using simulated data is shown in Fig. 7, where ML model outputs for training were derived from permeability results obtained through the Lattice-Boltzmann (LB) model. Proxy models are extensively utilized for rapid reservoir simulation, as traditional reservoir simulators can be computationally prohibitive for extensive parametric sensitivity analyses, particularly in complex physics systems involving composition simulations or intricate geological features (Zubarev, 2009; Al Haifi, 2019).

Another notable application of ML with simulated results (see Table 1) is demonstrated by Magzumov et al., 2021, where outputs from numerical and graphical solutions of the isothermal Buckley-Leverett equation for water injection were used to train a neural network model. This approach effectively leverages simulated data to enhance ML model training, providing a more efficient and practical method for predicting reservoir behavior.

These techniques showcase the benefits of using simulated results to create efficient ML models, facilitating rapid analysis and prediction while mitigating the computational burden associated with high-fidelity simulations.

#### 2.6.1.1. Smart Proxy Modeling Using Simulated Results.

Generative surrogate models, also known as Smart Proxy Models (SPM), extend the



**Fig. 14.** Workflow of traditional and smart proxy ML modeling using simulated results (Bahrami et al., 2022).

**Table 1**  
Simulated results from Buckley-Leverett equation used for training ML model in Magzymov et al., 2021.

Input parameters (7)	Input ranges	Output variables (3)
$M$	0.01 – 100	$\Lambda_{front}$
$S_{or}$	0.05 – 0.35	$S_{w\ front}$
$S_{wc}$	0.05 – 0.35	$p$
$k_{ro}^0$	0.1 – 1	
$k_{rw}^0$	0.1 – 1	
$n_o$	1.5 – 5	
$n_w$	1.5 – 5	

concept of traditional proxy models by incorporating generative approaches to enhance their functionality. In this context, generative models are employed to create new data points based on the patterns learned from existing simulated data. This approach allows the model to generate plausible scenarios or outcomes not explicitly included in the original training data, thus broadening its predictive capabilities.

For example, Nagao et al. (2024) employed generative surrogate models to tackle CO<sub>2</sub> sequestration challenges (see Fig. 15). Their approach begins by using a physics-based model to generate CO<sub>2</sub> onset time maps, capturing essential parameters such as pressure and temperature. To manage the complex and high-dimensional data, the input CO<sub>2</sub> plume propagation images are first compressed into a lower-dimensional latent space. This latent space is characterized by mean and variance values obtained through a Variational Autoencoder (VAE).

In the workflow, the VAE takes the mean and variance of these latent variables and samples them according to a normal distribution. This stochastic process ensures that the latent space captures a meaningful representation of the data. The VAE then decodes these latent variables back into the original image dimensions, generating predictions for CO<sub>2</sub> onset time maps.

Following this, a regression model is trained to predict the latent variables from monitoring data, which includes bottom-hole pressure (BHP) at the injector and distributed temperature sensing (DTS) data from monitoring wells. Once the regression model is developed, it acts as a generative model capable of estimating the latent space representation from new DTS and BHP data that were not part of the training set.

This latent representation is then used to reconstruct the higher-dimensional images, allowing the model to predict CO<sub>2</sub> onset time maps. By leveraging generative techniques, the model effectively incorporates uncertainties and improves predictive accuracy by generating additional plausible scenarios based on the learned latent space

representations.

This approach underscores the efficiency and adaptability of generative surrogate models in capturing complex phenomena and improving prediction accuracy. By generating new, plausible scenarios based on learned patterns, these models enhance their ability to analyze and predict real-world situations.

#### 2.6.2. ML models based residual of simulated results

The approach of improving a physics-based model's accuracy by stacking an additional model on top has garnered significant attention recently due to its straightforward application and potential for enhanced precision. This method involves the following workflow: First, results are generated from a physics-based model using a given input dataset. Next, the residuals—the differences between the actual observed results and the model's predictions—are computed. Subsequently, a machine learning (ML) or deep learning (DL) model is trained to predict these residuals based on the input dataset.

This technique effectively corrects the imperfections or inaccuracies inherent in the physics-based model by learning from its residual errors. For example, Molinari and Sankaran, 2021a utilized this approach to predict bottom hole pressure (BHP). They trained an ML model to estimate the residuals between measured BHP values and those calculated using multiphase steady-state flow equations, leveraging the input dataset matrix  $X$  for this purpose (see Fig. 16).

By focusing on the residuals, this method enhances the overall accuracy of the physics-based model, capturing and correcting the discrepancies that the original model might miss. This not only improves the model's predictions but also provides a more nuanced understanding of how well the physics-based model approximates real-world phenomena.

#### 2.6.3. Simulated-results based super-learner

Simulated-results based super-learners represent a sophisticated approach to integrating physics-based models with machine learning (ML) or deep learning (DL) techniques to enhance predictive accuracy and efficiency. This hybrid modeling strategy combines the rigorous, domain-specific insights of physics-based models with the adaptive, data-driven capabilities of ML/DL models. Here's a detailed breakdown of the two main types of super-learners.

- 1) Sequential or Series or Super Learner (shown in Fig. 17 (Sharma and Liu, 2022)) -In a sequential super-learner approach, the process is divided into distinct stages where the physics-based model and ML model interact in a sequence:

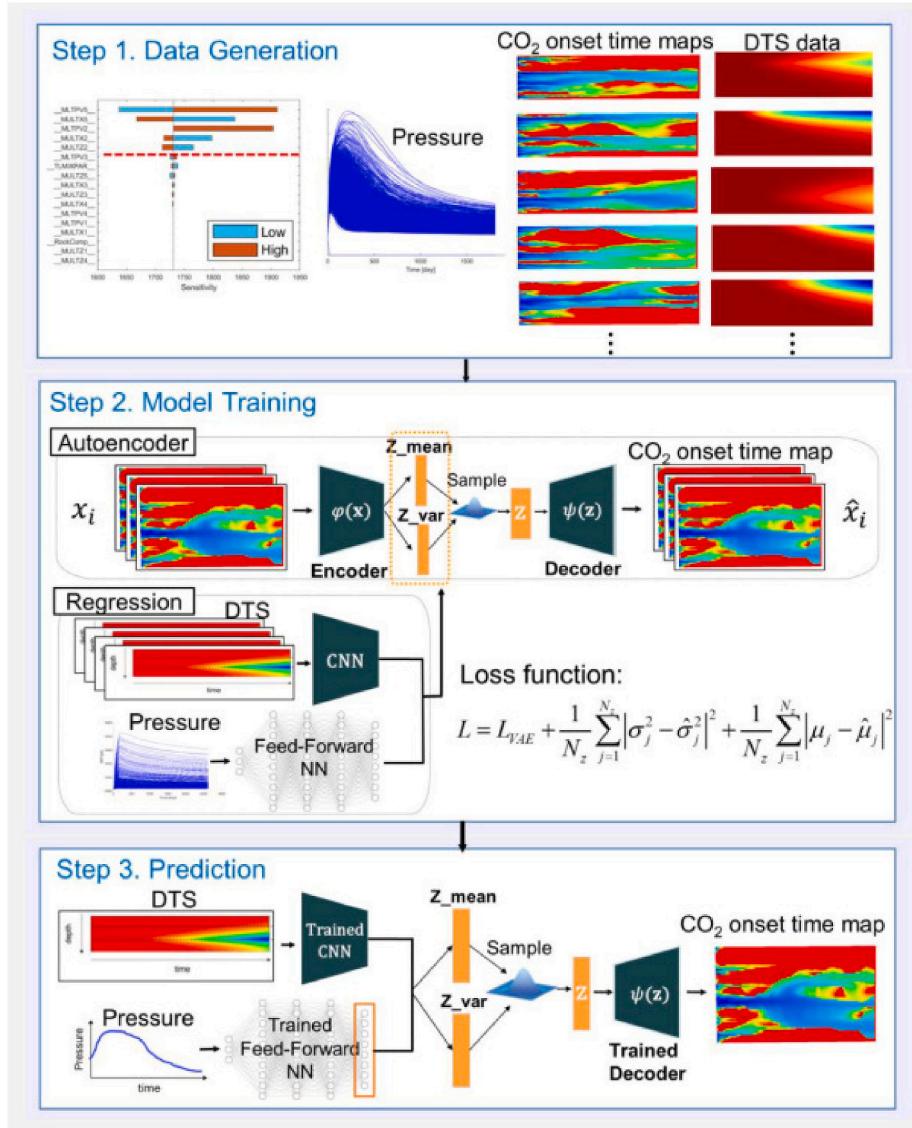


Fig. 15. Workflow of the generative proxy model used in Nagao et al., 2024.

$$X = [q_L \quad GLR \quad WCUT \quad P_{th}]$$

$$y = \varepsilon_{Phy} = \Delta P_{actual} - \Delta P_{Phy}$$

$$\Delta P_{pred} = \Delta P_{Phy} + y_{pred}$$

Fig. 16. Bottomhole pressure ( $y$ ) estimation using residual ML modeling (Molinari and Sankaran, 2021a).

#### Stage 1: Physics-Based Model:

Initially, a physics-based model, which is grounded in established scientific principles and equations, simulates the system or phenomenon of interest. This model produces outputs based on its inputs, which could include parameters like temperature, pressure, or composition, depending on the specific application.

#### Stage 2: Enhancing ML Inputs:

The results from the physics-based model are then used to augment

the input dataset for an ML model. Essentially, the output data from the physics-based model serves as additional features or variables for the ML model. This integration allows the ML model to learn from both direct data and the structured insights provided by the physics-based model.

#### Stage 3: ML Model Training and Parameter Estimation:

The ML model is trained using the combined dataset, which now includes outputs from the physics-based model. During training, the ML model can adjust or optimize parameters of the physics-based model based on patterns and relationships discovered in the data. This iterative refinement helps in improving the accuracy and reliability of the physics-based model's predictions.

This sequential approach effectively leverages the strengths of physics-based modeling for structured, principled simulation and the flexibility of ML for pattern recognition and parameter tuning. By combining these strengths, sequential super-learners can enhance predictive performance and provide more accurate insights.

- 2) In a parallel super-learner approach (shown in Fig. 18), the integration of physics-based and ML models occurs simultaneously, and their results are combined to produce the final prediction:

#### Stage 1: Independent Modeling

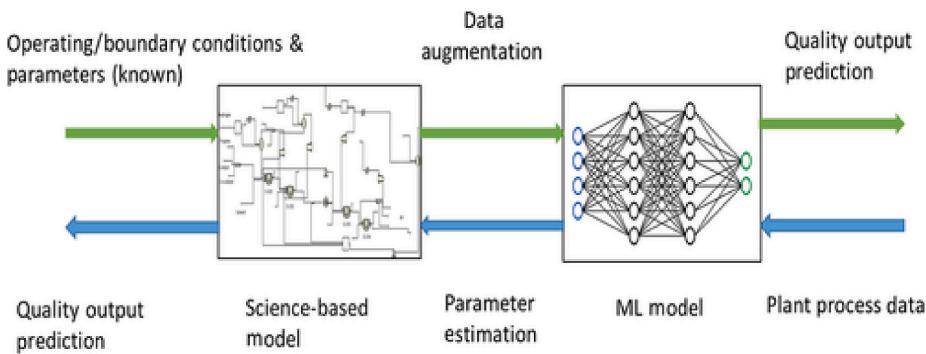


Fig. 17. Series physics-based super learner model workflow (Sharma and Liu, 2022).

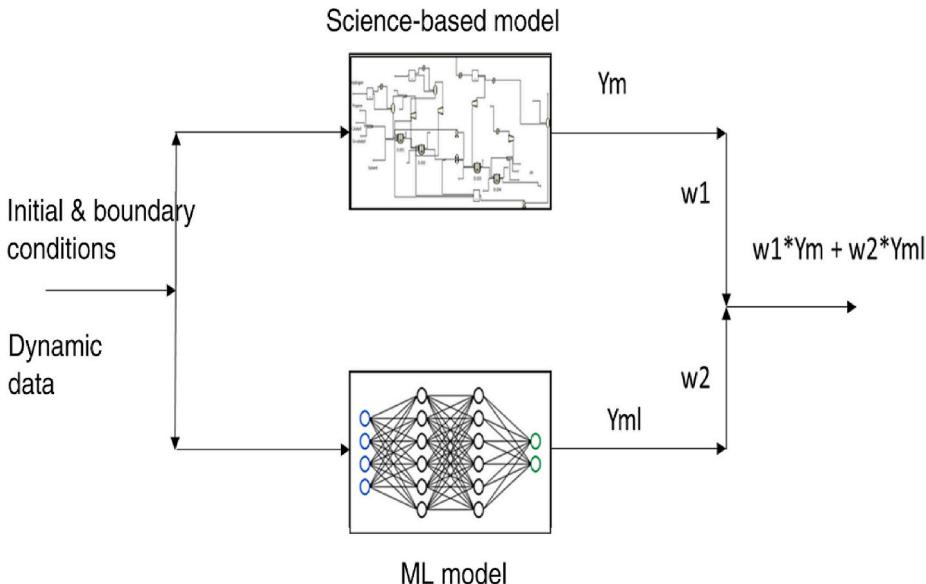


Fig. 18. Parallel physics-based super learner model workflow (Sharma and Liu, 2022).

Both the physics-based model and the ML model operate independently. The physics-based model generates predictions based on physical laws and principles, while the ML model makes predictions based on patterns and trends learned from historical or simulated data.

#### Stage 2: Combining Outputs

The outputs from both models are then combined using a combining function. This function could take several forms:

**Weighted Average:** Simple averaging where the results from each model are weighted according to their perceived reliability or importance.

**Linear Regression:** A regression model that uses the outputs from both models as inputs to predict the final outcome. This approach can capture linear relationships between the model results.

**ML Regression:** A more complex approach where another ML model combines the outputs from the physics-based and ML models. This method can capture non-linear relationships and interactions between the two sets of predictions.

The parallel super-learner approach benefits from the complementary strengths of physics-based and ML models. By combining predictions from both models, it can reduce individual model biases and leverage diverse sources of information, leading to more robust and accurate forecasts.

#### 2.7. Physics-regularized loss function

In machine learning (ML) regression, including simple linear

regression, the objective is to minimize the loss function of the form:

$$\text{Loss} = \text{Loss}^{\text{Accuracy}} + \alpha_1 \text{Loss}^{\text{Complexity}} \quad (\text{Eq.9})$$

where:

$\text{Loss}^{\text{Complexity}}$  is a loss due to regularization function on the model parameters to avoid overfitting (Like L1/L2 Regularization of weight parameter, to avoid super-high weight values).

$\text{Loss}^{\text{Accuracy}}$  is known as data loss, it could be measured in terms of mean square error (MSE) or absolute relative error (AE) and

$\alpha_1$  is a weight that determines the relative importance of the complexity term.

To ensure that the model's outputs adhere to physical laws, we can enhance the loss function with a physics-based regularization term:

$$\text{Loss} = \text{Loss}^{\text{Accuracy}} + \alpha_1 \text{Loss}^{\text{Complexity}} + \alpha_2 \text{Loss}^{\text{Physics}} \quad (\text{Eq.10})$$

$\alpha_2$  is a weight that balances the importance of the physics-based term relative to the other components.  $\text{Loss}^{\text{Physics}}$  could represent constraints derived from physical laws and/or underlying differential equations governing the system.

#### 2.7.1. Physics Informed Neural Networks

Physics-Informed Neural Networks (PINNs) exemplify the use of physics-regularized loss functions. PINNs incorporate physical laws into the training process of deep neural networks (DNNs) by integrating these laws into the loss function (as shown in Fig. 19). This approach

helps ensure that the learned model not only fits the data but also complies with underlying physical principles (Lagaris et al., 1998; Raissi et al., 2018).

Illustrative Example for PINNs is as follows:

Consider the quadratic function:

$$Y = ax^2 + bx + c \quad (\text{Eq.11})$$

the underlying ODE, initial conditions (ICs) are.

### 1. The Governing ODE

$$\frac{dY}{dx} = 2ax + b \quad (\text{Eq.12})$$

### 2. First Initial Condition IC:

$$Y(0) = c \quad (\text{Eq.13})$$

### 3. Second and Third ICs (Roots):

$$Y\left(\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}\right) = 0 \quad (\text{Eq.14})$$

The objective is to develop a Physics-Informed Neural Network (PINN) to approximate the quadratic function described in Eq. (11) without direct access to its explicit form. Instead, the model will be trained using the underlying physics governed by the ordinary differential equation (ODE) in Eq. (12), along with the initial conditions (ICs) specified in Eqs. (13) and (14), respectively. The PINN framework leverages these physical constraints to learn the solution in the absence of explicit data for  $Y$ .

For the given ODE, the first initial condition (Eq. (13)) is sufficient to obtain a unique solution. The additional conditions specified by the roots (Eq. (14)) are mathematically redundant but are included here to illustrate how multiple ICs (or boundary conditions, BCs, in the case of PDEs) can be incorporated into the PINN framework. This serves as a conceptual example of designing the loss function to include diverse physical constraints.

In real-world scenarios, we may encounter situations where.

1. Insufficient ICs or BCs: The problem does not provide enough conditions for a unique solution.
2. Data Losses: Measurements or experimental data are incorporated into the loss function as additional constraints.

The PINN framework must balance these multiple loss terms while ensuring adherence to the governing ODE or PDE, demonstrating its flexibility in addressing scenarios where explicit data is unavailable,

incomplete, indirectly accessible, or where the physical laws are highly nonlinear, coupled, or computationally expensive to solve directly.

If the explicit values of  $Y$  are unknown, a traditional data loss cannot be applied. Instead, the loss function for PINNs is constructed as follows:

$$\begin{aligned} \text{Loss} &= \text{Loss}^{\text{Complexity}} + \alpha_1 \text{Loss}_1^{\text{Physics}} + \alpha_2 \text{Loss}_2^{\text{Physics}} + \alpha_3 \text{Loss}_3^{\text{Physics}} \\ &\quad + \alpha_4 \text{Loss}_4^{\text{Physics}} \end{aligned} \quad (\text{Eq.15})$$

Let  $Y^{\text{Predicted}} = \text{NN}(x)$ , where  $\text{NN}(x)$  represents output of deep neural network with inputs  $x$  and take  $x = \text{any of the 100 random numbers from 0 to 1000}$ . Therefore, the Loss terms are defined as:

- $\text{Loss}_1^{\text{Physics}}$  - loss associated with the governing ordinary differential equation (ODE).

$$\text{Loss}_1^{\text{Physics}} = \text{ODE Loss} = \left( \frac{d\text{NN}(x)}{dx} - 2ax - b \right)^2 \quad (\text{Eq.16})$$

- $\text{Loss}_2^{\text{Physics}}$  - loss associated with first initial condition constraint.

$$\text{Loss}_2^{\text{Physics}} = \text{IC Constraint Loss 1} = (\text{NN}(0) - c)^2 \quad (\text{Eq.17})$$

- $\text{Loss}_3^{\text{Physics}}$  - loss associated with second initial condition or first root condition constraint.

$$\text{Loss}_3^{\text{Physics}} = \text{IC Constraint Loss 2} = \left( \text{NN}\left(\frac{-b - \sqrt{b^2 - 4ac}}{2a}\right) - 0 \right)^2 \quad (\text{Eq.18})$$

- $\text{Loss}_4^{\text{Physics}}$  - loss associated with third initial condition or second root condition constraint.

$$\text{Loss}_4^{\text{Physics}} = \text{IC Constraint Loss 3} = \left( \text{NN}\left(\frac{-b + \sqrt{b^2 - 4ac}}{2a}\right) - 0 \right)^2 \quad (\text{Eq.19})$$

Optimizing this loss function allows for the discovery of the underlying quadratic equation:

$Y = ax^2 + bx + c$  by solving the governing ODE and imposing the initial condition constraint.

If this problem were governed by a partial differential equation (PDE), the approach would remain similar, but instead of initial conditions, we would incorporate boundary conditions (BCs) to guide the network. For PDEs with multiple spatial variables, BCs specify the solution at the domain boundaries and can include Dirichlet (value-based), Neumann (derivative-based), or Robin (mixed) conditions. The

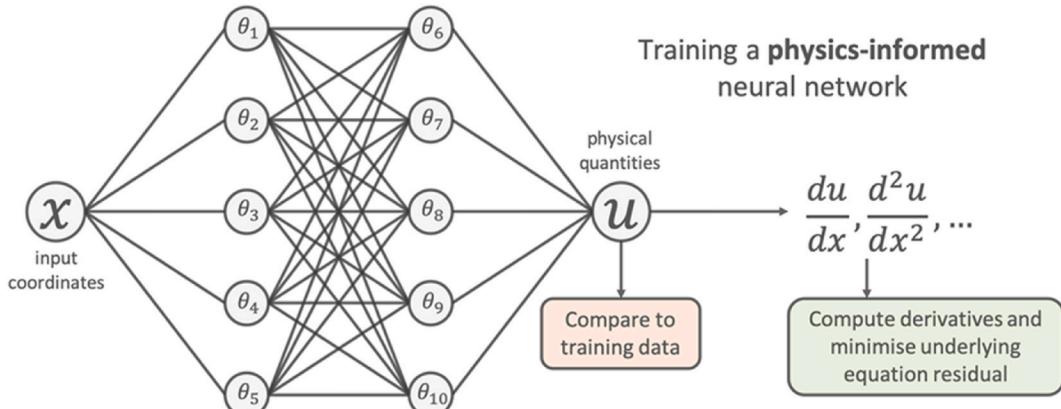


Fig. 19. Architect and training workflow of PINNs (Waheed, 2023).

corresponding loss terms for these BCs would be added to the total loss function, ensuring the solution adheres to the physical constraints at the boundaries.

By leveraging these physical constraints—whether from ODEs or PDEs—PINNs can effectively approximate the solution without requiring explicit data for the function being modeled. In reservoir engineering, PINNs can be applied to situations where the governing set of ODE/PDE is partially or entirely unknown, or where boundary and initial conditions (ICs and BCs) vary across different reservoirs or over time. For example, different governing equations may be required for the same reservoir if its operational conditions or physical properties change, such as transitioning from an early production stage to a water or gas drive phase. In such cases, the data loss term becomes crucial as it helps in addressing missing or uncertain information, while the physics-based constraints ensure that the model honors the underlying physical principles. By integrating both data loss and physics-based regularization, PINNs can effectively estimate unknown parameters and adapt to varying governing equations, leading to more accurate and reliable reservoir models.

Also, if the coefficient of a governing equation (ODE/PDE) is missing it could be easily discovered using PINN with the help of data loss (on sparsely available data)

$$\text{For example, } \frac{\partial Y}{\partial x} + \mu \frac{\partial^2 Y}{\partial t^2} = 3y \quad (\text{Eq. 18})$$

In cases where the coefficient ' $\mu$ ' is unknown, PINNs can estimate its value by iteratively adjusting ' $\mu$ ' to minimize the physics-based loss associated with the PDE while simultaneously minimizing the data loss on sparsely available training datasets. This dual optimization ensures that the learned solution aligns with both the governing physics and the available data.

This approach, referred to as data-driven ODE/PDE discovery, is particularly useful for identifying missing parameters or coefficients in complex systems. It has been successfully demonstrated in prior work, such as [Haghigat et al. \(2020\)](#), where PINNs were utilized to infer unknown physical parameters from limited data while preserving the underlying physical consistency of the solution.

The loss function for incorporating physics-based constraints in a machine learning model can be expressed as:

$$\begin{aligned} \text{Loss} = & \text{Loss}^{\text{Accuracy}} + \alpha_0 \text{Loss}^{\text{Complexity}} + \alpha_1 \text{Loss}_1^{\text{Physics}} + \alpha_2 \text{Loss}_2^{\text{Physics}} \dots \\ & + \alpha_n \text{Loss}_n^{\text{Physics}} \dots (\text{Eq. 19}) \end{aligned}$$

In this equation,

- $\text{Loss}^{\text{Accuracy}}$  represents data loss, such as mean squared error (MSE) or absolute error (AE), quantifying the difference between the actual and predicted values.
- $\text{Loss}^{\text{Complexity}}$  is a regularization term used to prevent overfitting by penalizing overly complex models. Regularization techniques such as L1 or L2 regularization are often used to ensure smoothness and simplicity of the learned function.
- $\text{Loss}_i^{\text{Physics}}$  represents the physics-based constraints incorporated into the model. These terms enforce the satisfaction of physical laws, including governing equations, initial conditions (ICs), and boundary conditions (BCs). Each physics-based term is weighted by  $\alpha_i$  which reflects its relative importance in the optimization process.

When not all initial conditions or boundary conditions are available,  $\text{Loss}_i^{\text{Physics}}$  helps ensure that the model adheres to known physical laws and constraints.

One recent application of Physics-Informed Neural Networks (PINNs) is detailed in [Wang et al. \(2024\)](#), where they use PINNs to predict adsorption isotherms for various shale cores. Their approach integrates Langmuir adsorption theory within a data-driven framework

to address the crucial need for accurate characterization of methane ( $\text{CH}_4$ ) and carbon dioxide ( $\text{CO}_2$ ) adsorption.

The PINNs model ([Wang et al., 2024](#)), as illustrated in [Fig. 20](#), employs automatic differentiation to incorporate physics-based knowledge directly into the neural network, which enhances predictive accuracy despite limited data availability:

**Left Sub-Network:** This component is designed to approximate the adsorption capacities for  $\text{CO}_2$  and  $\text{CH}_4$ .

**Right Sub-Network:** A simpler network focused on determining the constant terms in the Langmuir model, namely the maximum adsorption capacity ( $N_0$ ) and adsorption coefficients ( $K_1$  and  $K_2$ ), which represent physical properties related to adsorption.

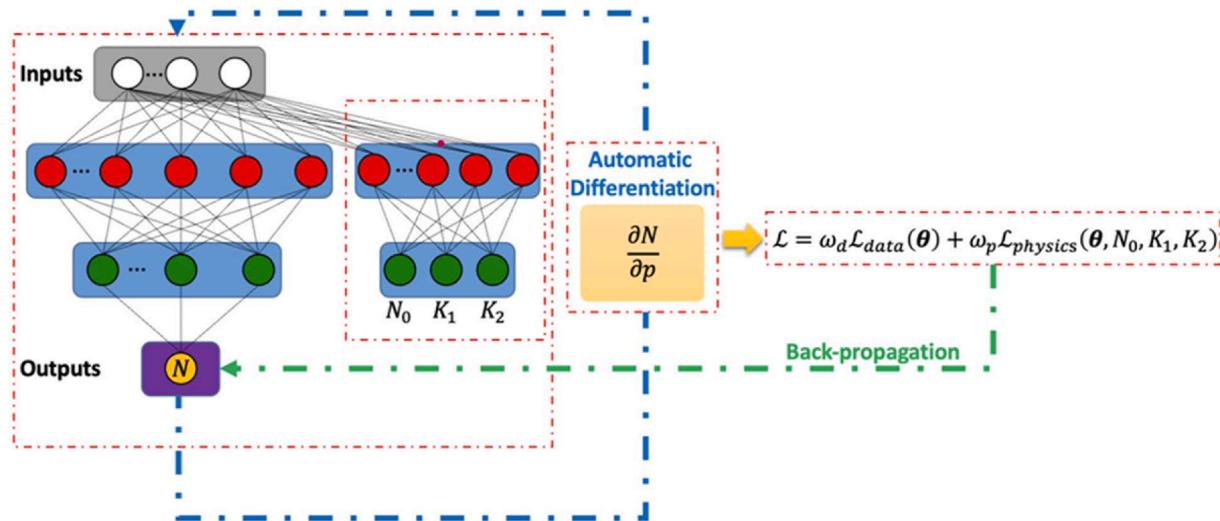
By including physics-based loss terms in the overall loss function, the model ensures adherence to governing equations, reducing the risk of unrealistic predictions. This method proves to be more effective than traditional machine learning models, especially in optimizing hydrocarbon recovery and understanding competitive adsorption phenomena.

Another example of this approach is illustrated in our recent work in [Sinha et al. \(2024\)](#), which takes a distinct approach by developing an augmented Physics-Informed Neural Networks (PINNs) model. Our model hybridizes the capacitance-resistance model (CRM) with non-linear regression capabilities, addressing specific challenges in solving the CRM ODEs with PINNs. Our method uniquely emphasizes building physics constraints from scratch and handling high-gradient issues in the loss function effectively.

[Sinha et al. \(2024\)](#), addresses some common challenges in machine learning models, particularly those trained on synthetic data or purely data-driven models that lack physical constraints. We compare these approaches with a focus on integrating physics-based constraints to improve model accuracy and prevent unphysical results. Our work identifies and addresses issues seen in existing models, such as excessive training weights and computational complexity, which can lead to numerical instability and longer training periods.

Our Augmented PINNs model is specifically designed to overcome these limitations by:

- Employing a custom neural network architecture with a limited number of layers and nodes to balance complexity and performance.
- Maintaining the original scale of the input data to preserve the physical relevance and clarity of the model.
- Utilizing finite difference methods for gradient computations instead of automatic differentiation ([Paszke et al., 2017; Baydin et al., 2018](#)), commonly used in python frameworks like TensorFlow and PyTorch. While automatic differentiation computes exact gradients with high precision via the chain rule, it often encounters stability issues with non-smooth functions or discontinuities, leading to large gradients and convergence difficulties. In contrast, the finite difference method enhances stability, offers flexibility in controlling time step sizes, and is computationally efficient. This approach bypasses back-propagation, reduces memory consumption, and is particularly suited for uniformly spaced temporal data, ensuring better control and convergence in complex optimization problems.
- Restricting the number of model parameters (weights, biases and unknown coefficients of the governing equations) relative to the size of the dataset to mitigate overfitting and improve the model's generalizability of the model. By ensuring that the model does not possess an excessive number of parameters in relation to the data, we prevent the potential trivial solution where the parameters simply mirror the dataset size. This promotes the extraction of physically meaningful patterns and facilitates the model's ability to generalize across unseen data, ensuring a more robust and physically relevant solution.



(a) PINNs Model Architecture

$$\mathcal{L}_{physics} = \left| \frac{\partial N}{\partial p} - \frac{N(K_1 y_1 + K_2 y_2)}{(1 + K_1 p y_1 + K_2 p y_2)^2} \right|^2$$

$$\mathcal{L}_{total} = \omega_d \mathcal{L}_{data}(\theta) + \omega_p \mathcal{L}_{physics}(\theta, N(\theta), K_1(\theta), K_2(\theta))$$

(b) Loss Function used in the PINNs

Fig. 20. Summary of the PINNs model used in Wang et al., 2024

### 3. Common areas of application of physics informed machine learning (PIML)

Table 2 presents a summary of common applications of physics-augmented machine learning models in the oil and gas industry, along with relevant examples.

### 4. Conclusions

This paper provides an extensive review of the application of machine learning (ML) and deep learning (DL) models in the oil and gas industry, with a focus on hybrid models that integrate physics-based constraints. By combining the adaptability of data-driven approaches with the structure of physics-based models, ML/DL frameworks offer greater flexibility and accuracy in reservoir modeling, production forecasting, and engineering tasks. This synergy between ML/DL and physics allows for more effective decision-making, particularly in handling complex systems and optimizing processes such as enhanced oil recovery, fluid phase equilibria predictions, and drilling operations.

The hybrid models, often referred to as physics-informed machine learning (PIML) models, address the limitations of both pure physics-based and traditional ML/DL methods. In physics-based models, the need for large datasets and high computational costs can hinder their real-time applicability. Meanwhile, ML models, without the guidance of physical laws, may generate unphysical or inaccurate results. Hybrid models bridge this gap by ensuring the data-driven models respect underlying physical laws while maintaining the flexibility of ML to handle complex relationships in the data.

The paper highlights key benefits of these hybrid models.

1. Improved Accuracy and Realism: By incorporating physical constraints, hybrid models prevent the generation of unphysical results that might arise from purely data-driven models.
2. Computational Efficiency: Compared to full physics-based simulations, hybrid models can reduce computational costs by using surrogate data-driven models while still retaining physical accuracy.
3. Enhanced Generalization: Hybrid models can generalize well to unseen data, making them valuable in real-time applications such as well performance prediction and enhanced oil recovery optimization.

However, the implementation of hybrid models also presents challenges. These include:

1. Balancing Complexity and Performance: Ensuring that the ML model remains computationally feasible while incorporating complex physical laws can be difficult, especially in high-dimensional systems.
2. Data Quality and Availability: Hybrid models depend heavily on high-quality, physically consistent data, and acquiring such data can be challenging, particularly for unconventional reservoirs.
3. Model Tuning and Validation: Hybrid models require extensive tuning to avoid overfitting and ensure that the physical constraints are not violated. The models must be validated against real-world data to ensure accuracy.

In conclusion, this paper demonstrates how hybrid ML models offer a promising path forward in petroleum engineering, combining the advantages of both physics-based and data-driven models. Future research should focus on refining these hybrid models, enhancing their ability to

**Table 2**

Summary of machine learning (ML) and deep learning (DL) model applications in the oil and gas industry.

Reservoir Engineering and Enhanced Oil Recovery	Drilling and Completions Engineering
<p>Surrogate reservoir models based on simulated data often incorporate physics-based loss functions and, in some cases, data-driven machine learning-augmented streamline-based modeling approaches (Samier et al., 2001) for flow diagnostics and optimization (Gaganis and Varotsis, 2012; Lino et al., 2017; Jeong et al., 2018; Navrátil et al., 2019; Bao et al., 2018; Guo et al., 2022; Liu et al., 2021; Hu et al., 2018). Machine learning and deep learning models, including Deep Neural Networks (DNNs), have been employed for modeling injector-producer relationships, predicting saturation distribution over time, and various enhanced oil recovery (EOR) applications, such as gas injection and carbon capture, utilization, and storage (CCUS) (Nwachukwu et al., 2018; Du et al. 2020; Liu et al., 2020; Yewgat et al., 2020; Hogg &amp; Yoon, 2020; He et al., 2021; Maniglio et al., 2021; Nagao et al., 2022; Thanh and Lee, 2022; Liu et al., 2020).</p> <p>Optimizing waterflood performance and well placement using machine learning techniques, such as clustering and other unsupervised methods, alongside reduced-order modeling approaches like POD and DMD. These methods help to efficiently analyze and predict reservoir behavior, enhancing decision-making and performance in reservoir management (Nikravesh et al., 2001; Wang et al., 2012; Hu et al., 2018; Liao and Tyagi, 2019; Liao et al., 2022; Zalavadia and Gildin, 2021).</p>	<p>Well path designing and construction using physics-based ML or DL methods. (Erge and Van Oort, 2022; Saini et al., 2022)</p> <p>Wellbore instability prediction (Geng and Wang, 2020).</p> <p>ROP optimized and tortuosity minimized by adjusting RPM, weight on the bit etc. (Hegde et al., 2015; Oyedere and Gray, 2020)</p>
<p>Petrophysics and Fluid Phase Equilibria</p>	<p>generalize while minimizing computational overhead, and exploring new methods for integrating physics into the model architecture more effectively.</p>
<p>Production Engineering</p>	<p><b>CRediT authorship contribution statement</b></p> <p><b>Utkarsh Sinha:</b> Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Validation, Visualization, Writing – original draft, Writing – review &amp; editing. <b>Birol Dindoruk:</b> Conceptualization, Data curation, Investigation, Methodology, Project administration, Supervision, Writing – review &amp; editing.</p> <p><b>Acknowledgments</b></p> <p>The authors would like to acknowledge help and support from members of the Joint Industry Project, “Interaction of Phase Behavior and Flow In Porous Media (IPBFPM)” Consortium and “Artificial Intelligence, Machine Learning, and Data Analytics for Energy Exploration and Production (AIM-DEEP)”, at University of Houston.</p> <p><b>Data availability</b></p> <p>No data was used for the research described in the article.</p> <p><b>References</b></p> <p>Ahmadi, M.A., Chen, Z., 2019. Machine learning models to predict bottom hole pressure in multi-phase flow in vertical oil production wells. <i>Can. J. Chem. Eng.</i> 97 (11), 2928–2940.</p> <p>Akhtar, I., Nayfeh, A.H., Ribbens, C.J., 2009. On the stability and extension of reduced-order Galerkin models in incompressible flows. <i>Theor. Comput. Fluid Dynam.</i> 23 (3), 213–237.</p> <p>Al Haifi, A.H.M., 2019. Confirmation of Data-Driven Reservoir Modeling Using Numerical Reservoir Simulation.</p> <p>Ali, S.F., 2003. Heavy oil—evermore mobile. <i>J. Petrol. Sci. Eng.</i> 37 (1–2), 5–9.</p> <p>Al sabaa, A., Gamal, H., Elkhataty, S., Abdelraouf, Y., 2022. Machine learning model for monitoring rheological properties of synthetic oil-based mud. <i>ACS Omega</i>.</p> <p>Aminzadeh, F., 2021. Application of Machine Learning in Reservoir Characterization. <i>Reservoir Characterization: Fundamentals and Applications</i>, pp. 487–523.</p> <p>Anifowose, F., Labadin, J., Abdulraheem, A., 2015. Improving the prediction of petroleum reservoir characterization with a stacked generalization ensemble model of support vector machines. <i>Appl. Soft Comput.</i> 26, 483–496.</p> <p>Araya-Polo, M., Alpak, F.O., Hunter, S., Hofmann, R., Saxena, N., 2018. Deep learning-driven pore-scale simulation for permeability estimation. In: <i>ECMOR XVI-16th European Conference on the Mathematics of Oil Recovery</i> (Vol. 2018, No. 1. European Association of Geoscientists &amp; Engineers, pp. 1–14. September.</p> <p>Aziz, K., 1979. <i>Petroleum Reservoir Simulation</i>, 476. Applied Science Publishers.</p> <p>Bahrami, P., Sahari Moghaddam, F., James, L.A., 2022. A review of proxy modeling highlighting applications for reservoir engineering. <i>Energies</i> 15 (14), 5247.</p> <p>Balabin, R.M., Lomakina, E.I., 2011. Support vector machine regression (SVR/LS-SVM)—an alternative to neural networks (ANN) for analytical chemistry? Comparison of nonlinear methods on near infrared (NIR) spectroscopy data. <i>Analyst</i> 136 (8), 1703–1712.</p> <p>Balaji, K., Rabiei, M., Suicmez, V., Canbaz, C.H., Agharzeyva, Z., Tek, S., et al., 2018. Status of data-driven methods and their applications in oil and gas industry. In: <i>SPE Europe Feature at 80th EAGE Conference and Exhibition</i>. OnePetro. June.</p> <p>Bao, A., Gildin, E., Zalavadia, H., 2018. Development of proxy models for reservoir simulation by sparsity promoting methods and machine learning techniques. In: <i>ECMOR XVI-16th. September.</i></p> <p>Baydin, A.G., Pearlmutter, B.A., Radul, A.A., Siskind, J.M., 2018. Automatic differentiation in machine learning: a survey. <i>J. Mach. Learn. Res.</i> 18 (153), 1–43.</p>



