PRODUCTION SCREENING AND OPTIMIZATION USING SMART PROXY MODELING

by

© Peyman Bahrami

A thesis submitted to the School of Graduate Studies

in partial fulfillment of the requirement for the degree of

Doctor of Philosophy

Faculty of Engineering and Applied Science

Memorial University of Newfoundland

September 2023

St. John's, Newfoundland

Dedication

I dedicate this dissertation in heartfelt tribute to my parents and my beloved spouse, Parisa, in recognition of their unwavering love and steadfast support.

Abstract

Numerical models are the primary tools to look into the fluid flow behavior in the complex and uncertain reservoir environment. Engineers use numerical models to perform crucial tasks in reservoir engineering, such as uncertainty quantification, history matching, production forecasting, and optimization, to eventually make the best decisions for field development. A conventional numerical model often consists of millions of grid blocks, and depending on the level of complexities within the model, it may take hours or days to perform a single run. A comprehensive study of a numerical reservoir model requires hundreds or thousands of repetitions, making the decision very costly and time-intensive. Proxy modeling is a solution for the computational cost related to the numerical models. They make a relationship between the input design parameters and the desired outputs by using various statistical/mathematical/data-driven underlying models. Nevertheless, they have their own limitations. The biggest disadvantage of the conventional proxy models is that they cannot keep the complexities within the reservoirs. It means they have no or limited sense of objects that exist in the reservoirs such as faults, boundaries, wells, etc.

The main objective of this research is to present smart proxy modeling (SPM) as a substitute for numerical models to address the computationally expensive and time-consuming drawbacks of numerical models and find a solution for keeping the complexities within the reservoir as conventional proxy models have. SPM is developed based on the implementation of pattern recognition and machine learning techniques, and it has an additional feature engineering step compared to the traditional known proxy models in the literature. The feature engineering step extracts new static and dynamic parameters from the numerical model. The constructed SPM takes only a few seconds to perform a single run. The SPM in this research is developed in grid-based and well-based types. The grid-based SPM can predict the grids’ properties, such as fluid saturation and pressure, and the well-based SPM is used to predict well production. Furthermore, the parallel implementation of the well-based SPM with grid-based SPM (named hybrid well-based SPM) is tested in this research.

The proposed SPM in this research is modified at different construction steps compared to existing SPMs in the literature that suffer from construction efficiency and reliability. Based on our literature review, we target our investigation into techniques to improve efficiency and accuracy by focusing on sampling, feature ranking, and underlying model construction. In existing SPM literature, only one technique is used during each construction step where there are opportunities to explore novel construction steps to improve overall SPM accuracy and efficiency. The presented sequential sampling technique avoids repeating the construction procedure from resampling and running the high-fidelity model, thereby saving time and making the SPM workflow more efficient. In the feature ranking step, an average of multiple ranking algorithms is used to find the best subset of input parameters which eventually helps the overall efficiency in the feature selection step. The performance of the convolutional neural network (CNN) as the underlying model is also tested and compared to the implemented artificial neural networks (ANN) in the literature.

In this research, the SPMs are constructed for two case studies. The first case study corresponds to a waterflooding scenario for the offshore Norway Volve field. The design parameters involve five parameters of the wells’ liquid production rates, and the objectives are to screen and optimize oil recovery. For the screening purpose, the grids’ pressure and oil saturation are considered as the outputs of the grid-based SPM. For production optimization, the wells' cumulative oil production is the output of the well-based SPM. Finally, the performance of well-based SPM coupled with two derivative-free optimizers, particle swarm optimization and genetic algorithm, are compared. The SPM with ANN underlying model provides an accuracy of 89-92% compared to the 94-99% of the CNN technique for the grid-based SPMs. However, for the well-based SPM, the goodness of fit for the 1D-CNN model is similar to the ANN model, but its accuracy (presented in MAPE) is slightly better than ANN. The well-based for this case study is coupled with PSO and GA optimization algorithms to find the best selection of designing parameters (individual well's LPR) and to maximize the cumulative oil production over ten years. Both optimizers are quite successful in finding the global optimum. Nevertheless, PSO shows a more reliable and faster convergence to the solution.

The second case study corresponds to a water alternating gas (WAG) scenario for the offshore Norway Norne field. This case study aims to test the whole procedure of SPM construction in another field with different levels of complexities and more design parameters. The design parameters for the WAG scenario are nine parameters of gas/water injection cycle, field gas/water injection rate, gas/water injection distribution between two injectors, and injectors' BHPs. Similar to the first case study, screening and oil recovery optimization are the targets for this case study. The trained CNN models give an accuracy of 85-87% for different timesteps of the grid-based dataset at the blind test. However, after adding five more sample points using the sequential LHS, the accuracy increases to 94-99%. The well-based SPM, similar to the first case study, does not give promising improvement in terms of accuracy.

Acknowledgments

First and foremost, I am incredibly grateful to my supervisor, Professor Lesley A. James, for her invaluable advice, continuous support, and patience during my Ph.D. study. Her immense knowledge and plentiful experience have encouraged me throughout my academic research and daily life.

I would like to express my sincere gratitude to my supervisory committee, Dr. Sohrab Zendehboudi, and Dr. Ray Gosine, for their valuable feedback on my thesis. I would also like to thank my internal referees, Dr. Ronald D. Haynes and Dr. Amer Aborig, and my external referee, Dr. Ashkan Jahanbani Ghahfarokhi for accepting to review the dissertation.

I must express my thank and gratitude to Norah Hyndman for her feedback on the research. I would also like to give special thanks to Dr. Masoud Asadollahi, and Dr. Pezhman Kazemi for their technical support and feedback on this work.

I would like to thank the Hibernia Management and Development Company (HMDC), Chevron Canada Ltd., Energy Research and Innovation Newfoundland and Labrador (ERINL), the Natural Sciences and Engineering Research Council of Canada (NSERC), the Province of Newfoundland and Labrador, and Mitacs for financial support. I acknowledge Equinor for sharing the reservoir model and data publicly.

Words cannot express my thankfulness for the support and encouragement provided by my lovely wife, Parisa, during my studies.

Table of Contents

List of Figures

List of Tables

Abbreviations

ACO	ant colony optimization
ANFIS	adaptive neuro fuzzy inference system
ANN	artificial neural networks
BBD	box-Behnken design
BHP	bottom hole pressure
CCD	central composite design
CNN	convolutional neural networks
CRM	capacitance-resistance modeling
DE	differential evolution
DEIM	discrete empirical interpolation method
GA	genetic algorithm
GAGD	gas-assisted gravity drainage
GBM	gradient boosting machine
GP	genetic programming
GSA	global sensitivity analysis
KG	kriging
LHS	Latin hypercube sampling
LSA	local sensitivity analysis
LSTM	long short-term memory
MAPE	mean absolute percentage error
MARS	multivariate adaptive regression splines
MC	Monte Carlo
MCMC	Markov chain Monte Carlo
MFM	multi-fidelity model
NPV	net present value
OAS	orthogonal array sampling
PBD	Plackett-Burman design
PCE	polynomial chaos expansion
PM	proxy model
POD	proper orthogonal decompositions

PR	polynomial regression
PSO	particle swarm optimization
RBF	radial basis functions
ReLU	rectified linear unit
RF	random forest
RNN	recurrent neural networks
ROM	reduced-order model
RSM	response surface model
SAGD	steam-assisted gravity drainage
SBO	surrogate-model based optimization
SPM	smart proxy model
SVM	support vector machine
SVR	support vector regression
TPM	traditional proxy model
TPWL	trajectory-piecewise linear
WAG	water alternating gas
XGboost	extreme gradient boosting

Chapter 1. Introduction

1.1. Problem Statement and Motivation

The most crucial role of a reservoir engineer is to plan the right strategy for managing and developing a field. Any decisions on field development plans can affect the volume of produced oil, cost, and revenue. This is essential because it directly impacts the economic viability and overall success of oil and gas operations. The subsurface environment is inherently unpredictable, with various geological and geophysical factors at play. This complexity makes decision-making a hard task. Decisions are even more complicated to make when other constraints are involved, such as resources and operation limitations. Resource constraints, including available manpower and equipment, as well as operational limitations, like safety regulations, add layers of complexity to the decision-making process. These factors necessitate a comprehensive approach to field development. Numerical models can be a handy tool to help engineers to simulate different scenarios and evaluate their impact on field performance. They provide a systematic way to account for all relevant constraints and variables in the decision-making process. Numerical models help solving complicated problems by breaking them down into smaller grids and using specific mathematical rules. Their accuracy depends on how small we select these grids. If we make the cells very small, we get highly accurate results, but it takes a long time to do the calculations. On the other hand, if we use larger grids, it's faster, but the results might not be accurate.

Numerical models help assess field production and find new opportunities in the field. However, they come with their own set of challenges, such as complexity of handling numerous parameters and strategies, as well as the significant time investment required for each simulation run. These challenges can lead to simplifications in production optimization and hinder the identification of the best solution. These models with the help of advanced computer technologies run faster than before, but engineers still tend to get more detailed information from the subsurface environment by considering smaller grids and removing the simplifying assumptions. In industries like oil and gas, where data is collected rapidly, understanding subsurface conditions is crucial. However, updating plans quickly and accurately can be difficult using numerical models. This has led to the exploration of computationally efficient proxy models (PMs) as an alternative technique.

PMs are representations of numerical models. They can deal with some of the challenges, and they are mainly beneficial where the computational burden associated with simulators, particularly in cases where running multiple simulations is resource-intensive [1]. Conventional proxy models use the underlying statistical, mathematical and data-driven models to create a simple relationship between input design variables and the output. PMs can be applied to various reservoir engineering tasks, such as production forecasting and optimization [2], uncertainty analysis [3], and history matching [4]. In order to build a PM, it is necessary to select the influential input parameters on the desired output under investigation. This demands a deep understanding of the problem at hand and the ability to recognize which factors are most influential. [5]. Proxy models are built by running numerical simulations with carefully chosen input parameters and recording corresponding output data. However, due to practical constraints, proxy models often simplify the representation by omitting some input parameters related to geology, geophysics, fluid properties, and rock characteristics. While this simplification reduces computational time, it can also affect the accuracy compared to full numerical models.

PMs can be categorized based on different aspects such as time dependency, approximation strategy, or objective. Various categorization techniques for PMs are elaborated in Chapter 2, but in general, some of the main classes include multi-fidelity, reduced-order, and data-fit models.

Multi-fidelity PMs utilize physics rules to create a simplified version of the main model. Reduced-order PMs aim to eliminate irrelevant parameters by projecting the problem into a lower-dimensional space, while data-fit models employ statistical and data-driven techniques to replicate the model for output prediction. All these categories of PMs construct simpler models to predict the output by establishing a direct relationship between input design parameters and the output. In this work, a novel classification of PMs that categorizes them into four groups: multi-fidelity, reduced-order, traditional, and smart proxy models, based on their development strategy, is elaborated in Section 2.2. Smart proxy modeling (SPM), used in this work, is a type of PM with additional steps in development.

Smart proxy modeling takes proxy modeling a step further by incorporating advanced techniques, such as pattern recognition and feature engineering, into its development process. These techniques enable SPMs to uncover previously unseen relationships between input and output parameters that conventional PMs may miss. This is achieved by extracting a wide range of both static and dynamic parameters from simulation runs, allowing SPMs to capture unseen relationships. The methodology of developing an SPM was first introduced by Mohaghegh in 2006 [6], but it was recognized with the term “surrogate model” at that time. A few years later, in 2018, he used the term “smart proxy model” for the rest of his work [7]. The time needed to run an already constructed SPM is just a few seconds, similar to the PM. However, SPMs surpass conventional PMs in accuracy due to their incorporation of additional parameters into the model. The SPM can be constructed in two different types of grid-based and well-based. The grid-based SPM predicts the outputs at grid level such as grids’ saturations and pressure, while the well-based SPM predicts outputs at well level such as oil production for the wells. The results obtained from screening with grid-based SPM can be used for various purposes such as well placement optimization to increase oil production or observing pressure changes within the reservoir. Monitoring pressure is particularly crucial in immiscible gas injection recovery methods to ensure that the pressure remains below the minimum miscibility pressure.

The main motivation behind this research project, or any SPM, is to provide an efficient alternative to the time-intensive numerical models that works as a fast tool for production screening and optimization of a field. Furthermore, we are motivated to provide a comprehensive guideline to familiarize readers with the current state of SPM, its application in literature, and the different steps required to construct the SPM. We also investigate novel techniques in the development stages (such as sampling, feature extraction, and underlying models) to further improve the SPM capability compared to the existing literature.

1.2. Research Objectives

This work aims to develop SPMs to screen and optimize production during waterflooding and water alternating gas (WAG) processes. To perform screening, grid-based SPMs are developed from scratch to predict the grids' outputs, such as fluid saturations and pressure. For the production optimization tasks, well-based SPMs are developed to predict the cumulative oil production of the wells and the field. Furthermore, the efficiency and accuracy of the proposed SPMs are improved in various steps such as sampling, feature engineering, feature ranking, feature selection, and underlying model. The main research objectives of this study are as follows:

- Identify and fulfill existing gaps in PM and SPM classification techniques.
- Describe PM and SPM development workflow, and investigate the main existing techniques for the individual steps.

- Review existing work on the subject of PM and SPM to get familiar with their applications in reservoir engineering.
- Construct grid-based SPMs to screen a waterflooding case in terms of grids' oil saturation and pressure involving five design parameters.
- Construct a hybrid well-based SPM that works in parallel with grid-based SPM to predict cumulative oil production in the same waterflooding case study.
- Compare the performance of different derivative-free algorithms when coupled with the well-based SPM to find the optimal design parameters.
- Repeat the whole procedure with another case study that has different levels of complexities and consider more design parameters with another EOR process (WAG case study).

1.3. Novelties and Contributions

In existing SPM literature [8]–[13], only one technique for the individual steps during development is tested. We identify this as a gap in literature where SPM models can be improved through a more rigorous sensitivity analysis of method selection on the overall SPM accuracy. The detailed novelties and contributions implemented in this research are as follows:

- A new classification of PMs, including multi-fidelity models (MFM), reduced-order models (ROM), traditional proxy models (TPM), and SPM is proposed in the literature review. The existing classification in the literature could not cover all proxy models [7], [14]–[20], and in some studies, different proxy models are even considered as one model [21], [22].
- Sequential sampling technique is introduced. Sequential sampling helps to construct the SPM with the lowest number of high-fidelity model execution, and it avoids resampling,

thereby saving time and making the SPM workflow more efficient. In existing SPM literature [8]–[13], only stationary Latin hypercube sampling technique is implemented.

- The average feature ranking technique is applied to find the best prioritization of input parameters. It provides a confident ranking for the feature selection step. In the feature selection step then, the best subset of input parameters can be identified.
- Convolutional neural network (CNN) deep learning modeling is trained and compared to ANN underlying model. In the literature, the majority of the studies only investigates the implementation of ANN as the underlying model of the SPMs. As a result, the need for testing novel models such as deep learning techniques is required.
- A new tiering system around the wells is introduced. This tiering system includes the grids surrounding the well at various radius for the whole well trajectories. New features can then be engineered based on the included grids in the tiers. In the existing literature, a tiering system considers the grids based on the drainage area of the wells [8], [9]
- A hybrid well-based SPM is constructed to consider the effect of surrounding grids' dynamic parameters on the individual wells' production. The dynamic parameters are obtained in the parallel implementation of the grid-based and well-based SPMs.
- The performance of two derivative-free optimizers, particle swarm optimization (PSO) and genetic algorithm (GA), when coupled to well-based SPM, are investigated. The two optimizers help find the best design parameters selection and recover more oil.

1.4. Outline of Thesis

The thesis includes six chapters in total. The organization of the thesis is as follows:

Chapter 1 is the current chapter and presents the thesis problem, motivations, objectives, and novelties.

Chapter 2 provides a literature review on the PM and SPM, including the classifications, workflow description, and applications in reservoir engineering. A similar version of this chapter is published in Energies journals.

Reference: P. Bahrami, F. Sahari Moghaddam, and L. A. James, “A review of proxy modeling highlighting applications for reservoir engineering,” *Energies*, vol. 15, no. 14, Art. no. 14, Jan. 2022, doi: 10.3390/en15145247.

Chapter 3 presents the implementation of a grid-based SPM to screen waterflooding in the Volve case study. The SPM is constructed by considering five design parameters. A similar version of this chapter is published in Geoenergy Science and Engineering journal.

Reference: P. Bahrami and L. A. James, “Screening of waterflooding using smart proxy model coupled with deep convolutional neural network”, *Geoenergy Science and Engineering*, vol. 221, p. 111300, Feb. 2023, doi: 10.1016/j.petrol.2022.111300.

Chapter 4 introduces a well-based SPM to predict and optimize the production during waterflooding in the Volve case study. The SPM is constructed by considering five design parameters. A similar version of this chapter has been presented at SPE conference.

Reference: P. Bahrami and L. A. James, “Field Production Optimization Using Smart Proxy Modeling; Implementation of Sequential Sampling, Average Feature Ranking, and Convolutional Neural Network”, Presented at 23 SPE Canadian Energy Technology Conference and Exhibition, Calgary, Canada, March 2023, SPE-212809-MS, https://doi.org/10.2118/212809-MS

Chapter 5 describes the constructed grid-based and well-based SPMs for screening and production optimization purposes in a WAG process. The Norne field case and a total of nine design parameters are considered for this investigation. This chapter is under preparation for journal submission.

Chapter 6 provides conclusions for the thesis research and summarizes its contributions. In addition, recommendations are made for future research work on the SPM.

Chapter 2. A Review of Proxy Modeling Highlighting Applications for Reservoir Engineering

Preface

A version of this chapter has been published in the Energies journal. I am the primary author, along with the co-authors, Farzan Sahari Moghaddam, and Dr. Lesley A. James. The detail of the author's contributions are as follow:

Peyman Bahrami: Conceptualization, Methodology, Writing–original draft, Writing–Review & Editing, and Visualization

Farzan Sahari Moghaddam: Writing–Review & Editing

Dr. Lesley A. James: Conceptualization, Methodology, Writing–Review & Editing, and Supervision

Reference: P. Bahrami, F. Sahari Moghaddam, and L. A. James, "A review of proxy modeling highlighting applications for reservoir engineering," *Energies*, vol. 15, no. 14, Art. no. 14, Jan. 2022, doi: 10.3390/en15145247.

Abstract

Numerical models can be used for many purposes in oil and gas engineering, such as production optimization and forecasting, uncertainty analysis, history matching, and risk assessment. However, subsurface problems are complex and non-linear, and making reliable decisions in reservoir management requires substantial computational effort. Proxy models have gained much attention in recent years. They are advanced non-linear interpolation tables that can approximate complex models and alleviate computational effort. Proxy models are constructed by running high-fidelity models to gather the necessary data to create the proxy model. Once constructed,

they can be a great choice for different tasks such as uncertainty analysis, optimization, forecasting, etc. The application of proxy modeling in oil and gas industry has had an increasing trend in recent years, and there is no consensus on the correct choice of proxy model. As a result, it is crucial to better understand the advantages and disadvantages of various proxy models. The existing work in the literature does not comprehensively cover all proxy model types, and there is a considerable requirement for fulfilling the existing gaps in summarizing the classification techniques with their applications. We propose a novel categorization method covering all proxy model types. This review paper provides a more comprehensive guideline on comparing and developing a proxy model compared to the existing literature. Furthermore, we point out the advantages of smart proxy models (SPM) compared to traditional proxy models (TPM) and suggest how we may further improve SPM accuracy where the literature is limited. This review paper first introduces proxy models and shows how they are classified in the literature. Then, it explains that the current classifications cannot cover all types of proxy models and proposes a novel categorization based on various development strategies. This new categorization includes four groups multi-fidelity models (MFM), reduced-order models (ROM), TPM, and SPM. MFMs are constructed based on simplifying physics assumptions (e.g., coarser discretization), and ROMs are based on dimensional reduction (i.e., neglecting irrelevant parameters). Developing these two models requires an in-depth knowledge of the problem. In contrast, TPMs and novel SPMs require less effort. In other words, they do not solve the complex underlying mathematical equations of the problem; instead, they decouple the mathematical equations into a numeric dataset and train statistical/AI-driven models on the dataset. Nevertheless, SPMs implement feature engineering techniques (i.e., generating new parameters) for its development and can capture the complexities within the reservoir, such as the constraints and characteristics

of the grids. The newly introduced parameters can help find the hidden patterns within the parameters, which eventually increase the accuracy of SPMs compared to the TPMs. This review highlights the superiority of SPM over traditional statistical/AI-based proxy models. Finally, the application of various proxy models in the oil and gas industry, especially in subsurface modeling with a set of real examples, is presented. The introduced guideline in this review aids the researchers in obtaining valuable information on the current state of PM problems in the oil and gas industry.

Keywords: Proxy model; Surrogate model; Traditional proxy; Smart proxy; Multi-fidelity; Reduced-order; Sensitivity analysis; Sampling; Machine learning; Application

2.1. Introduction

In the late 1990s, with the increase in the computational power of computers, industries increased the use of numerical models to solve complex problems. Numerical modeling is a mathematical representation of physical or chemical behaviors wherein the governing properties in the process are spatially and temporally characterized [23]. It plays a significant role in the development, uncertainty analysis, and optimization of many processes in various areas such as engineering, geology, geophysics, applied mathematics, and physics. Numerical models can reduce time and cost compared to more traditional trial and error methods [24]. Nevertheless, achieving accurate results quickly has always been a challenge, even using numerical models or software implementing them. Numerical models divide the problem into a large number of small cells and solve it based on discrete calculus, considering the initial conditions, boundary conditions, and underlying assumptions [25]. The accuracy of a numerical model depends on the size of the cells used to capture the governing equations of the problem or grid spacing. A fine-grid numerical model is also referred to as a high-fidelity model [26]. There is always a trade-off between the accuracy and speed of numerical models. Performing an analysis with a low number of cells might be quick; however, it sacrifices the quality of the results, or it does not yield convergence. Conversely, a high number of cells increases the computational time, so obtaining the results at the various realizations of the problem is very time-consuming [27]. In recent years, improvements in computational hardware and software, and the emergence of the parallel processing of CPUs have boosted the speed of running numerical models. However, as computers become more powerful, users, in turn, are demanding more, such as applying more parameters or removing simplifying assumptions, in order to increase the quality of the results. Therefore, the availability of computing resources remains a limiting factor, and researchers are looking for ways to reduce the computational load related to the use of numerical models or the software implementing them.

In the oil and gas industry, and especially in reservoir modeling, there are many sources of data, such as drilling, seismic, well tests, production, etc., that are collected very quickly, which may change the understanding of subsurface conditions and uncertainties. In parallel, field development plans need to be updated in shorter periods, and performing real-time analysis can be very beneficial to understanding the evolving conditions in the reservoir. However, having a real-time analysis limits the usage of these expensive numerical models, or the software implementing them. As a result, the application of computationally efficient proxy models (PMs) has been investigated in recent years.

PMs, also called surrogate models, or metamodels, are substitutes or approximations of numerical models, mathematical models, a combination of them (such as models behind a complex software), or even an experimental test. A simple description of proxy models is that they are advanced interpolation tables from which we can quickly interpolate ranges of non-linear data to find an approximate solution. Figure 1 demonstrates the use of these equivalent terms in the literature extracted from the Web of Science Core Collection by searching the exact keywords in the titles of the papers published since 1990 [28]. As shown in this figure, there has been an increasing trend in the use of these models since 2000, and "surrogate modeling" is the most widely applied term in the literature. In this paper, the term "proxy modeling" has been selected and will be used henceforth. Additionally, "high-fidelity model" will be used to describe the model (numerical, mathematical, or a combination) that the PM is trying to approximate.

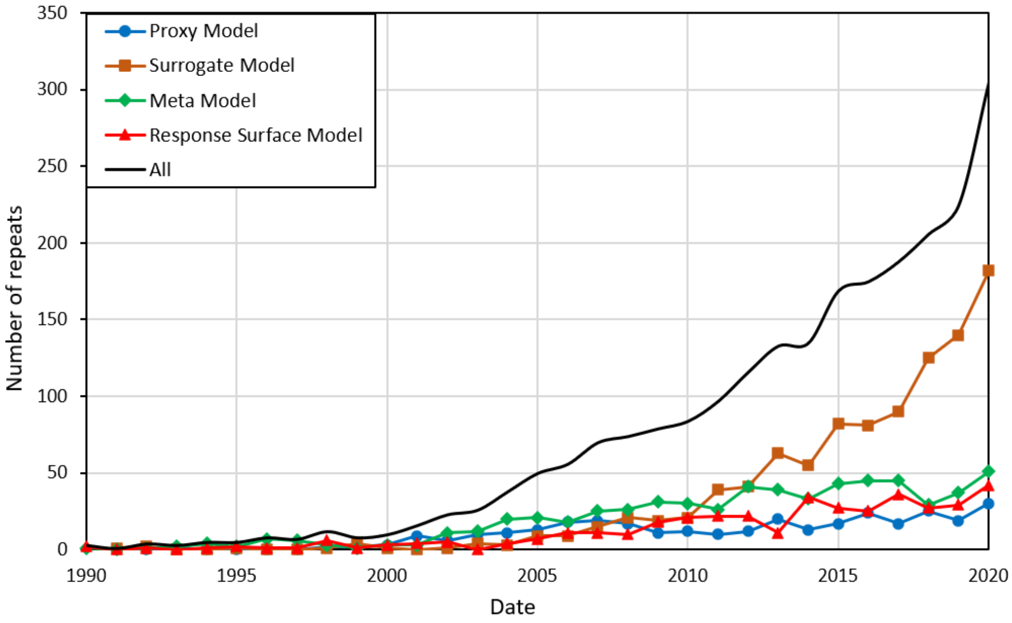


Figure 1. The use of equivalent terminologies for "proxy model" in the literature extracted from the Web of Science Core Collection since 1990 [28]

In proxy modeling, a modest sample of input parameters is chosen, and the high-fidelity model is run within the given space of the parameters to obtain the outputs. Then, the PM fits these data. This PM is only valid for the given set of inputs and corresponding search spaces. The advantage of a PM is that once it is developed, it only requires a few seconds to run. PMs provide the increased speed required for decision-making compared to high-fidelity models; however, the accuracy of the models remains a challenge. It should be noted that the advantage of using a PM is its high speed, and that a high-fidelity model still provides the most accurate results over all the spatial and temporal locations.

There are different objectives for using PMs, including sensitivity analysis, uncertainty quantification, risk analysis, and optimization [29]. This review paper highlights the use of proxy modeling in the oil and gas industry, in particular, reservoir modeling and related areas such as history matching, field development planning, and reservoir characterization. Forrester et al. [29] discuss four common applications of PMs: (1) providing accelerated results from expensive high-fidelity models such as a software; (2) calibration mechanisms for predictive models with limited accuracy; (3) dealing with noisy or missing data; and (4) gaining insight into the functional relationships between parameters. It must be remembered that PMs utilize and boost the usage of high-fidelity models by creating an approximation, and achieving the objectives still requires the implementation of the high-fidelity models as the initial and main step in the proxy modeling development process.

This review aims to provide a set of guidelines for PM development by introducing the different classes of PM, the methodology, and their applications in oil and gas. In Section 2.2, different classes of PMs are reviewed, as well as the proposed classification in this paper; the steps to create a PM are explained in Section 2.3; and the application of PMs in oil and gas engineering is discussed in Section 2.4.

In this work, we provide a new classification of PMs, including multi-fidelity models (MFMs), reduced-order models (ROMs), traditional proxy models (TPMs), and smart proxy models (SPMs). The existing classification in the literature could not cover all proxy models [7], [14]–[20], and in some studies, different proxy models are even considered as one model [21], [22]. To fully comprehend the models, the advantages and disadvantages of each class are discussed.

The superiority of fast-to-construct TPMs and SPMs compared to other time-consuming models (MFM and ROM) is demonstrated. Both TPMs and SPMs require less scientific knowledge of the problems because they decouple the mathematical equations of the problem into a numeric dataset. However, SPMs preserve the complexities within the reservoir, such as the faults, boundaries, and characteristics of the grids, compared to TPMs. As a result, SPMs provide higher accuracy by considering more related parameters within the reservoir. In this paper, we discuss more thoroughly the methodology to construct an SPM, describing in detail the different steps, such as sampling and training the underlying model. In the existing SPM literature [8]–[13], only one technique for each step is tested. For example, only one type of underlying model (ANN) or one type of sampling technique (Latin hypercube sampling) is tested. We identify this as a gap in the literature where SPM models can be improved through a more rigorous sensitivity analysis of method selection on the overall SPM accuracy.

2.2. Proxy Modeling Classification

PMs can be categorized in various ways, such as by their objective/application, the approximation strategy used, or their time-dependency. Figure 2 presents the various ways PMs are classified in the literature.

Ahmed and Qin [14] divided PMs into two groups, black-box-based and physics-based approaches, according to the approximation strategy. In a black-box-based approach, the high-fidelity model cannot be modified, and the PM makes a less expensive approximation of the relationship between inputs and outputs. Conversely, a physics-based approach modifies the governing equations of the problem to make it computationally cheaper, which will be discussed later in this section. Black-box PMs are further divided into parametric and nonparametric, according to the nature of the unknown parameters. In both parametric and nonparametric models, the output parameters are determined using the initial training set; however, parametric models do not implement the training set to make the predictions, whereas nonparametric models do [30]. Polynomial regression (PR) is an example of a parametric model, and kriging (KG), artificial neural networks (ANN), radial basis functions (RBF), multivariate adaptive regression splines (MARS), and support vector regression (SVR) are examples of non-parametric approaches.

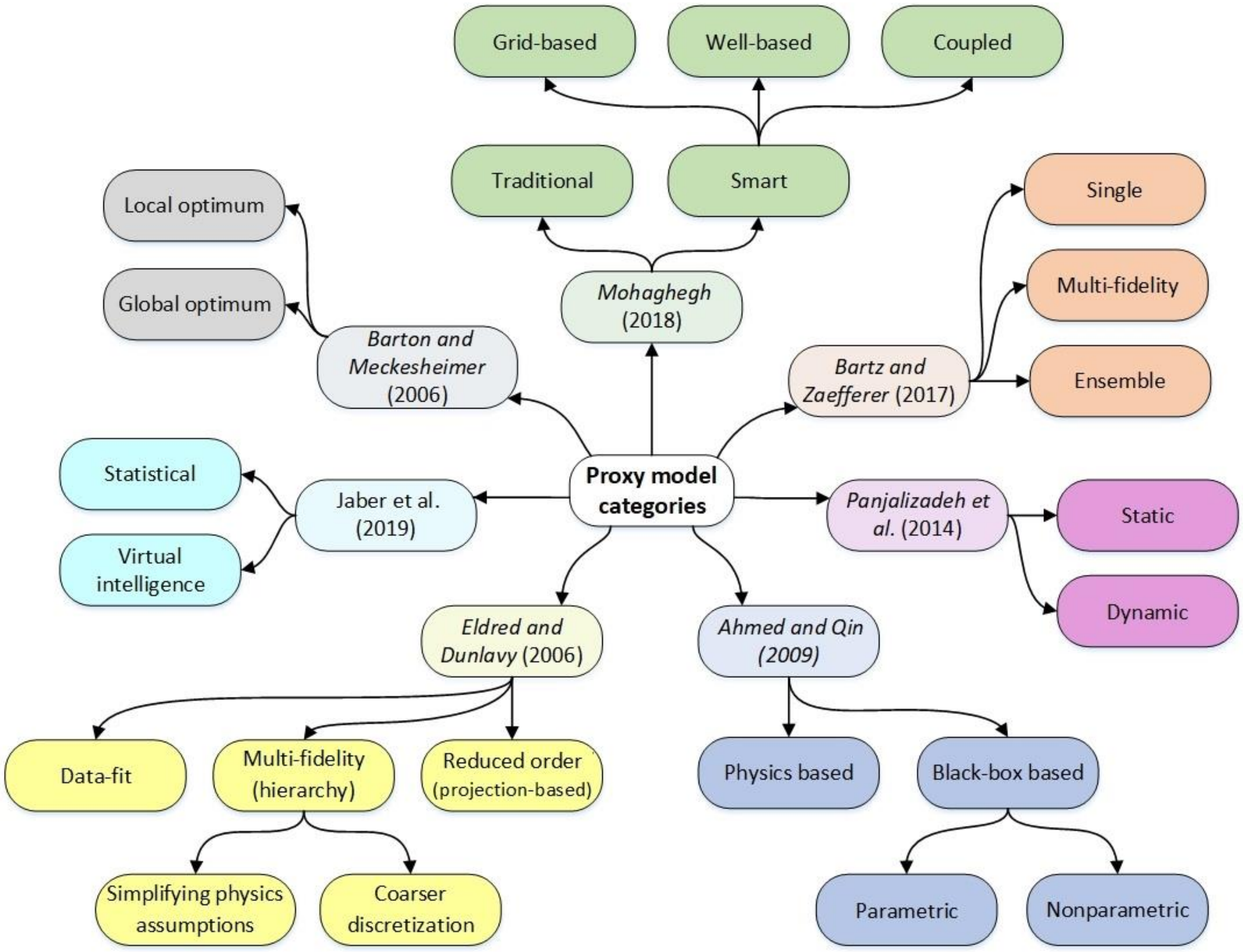


Figure 2. Summary of categorization classes in the literature for proxy models (Ahmed and Qin [14], Eldred and Dunlavy [15], Panjalizadeh et al. [16], Mohaghegh [7], Bartz and Zaefferer [17], Barton and Meckesheimer [18], Jaber et al. [19])

Eldred and Dunlavy [15] classified PMs into data-fit, multi-fidelity (also called hierarchy), and reduced-order types (also called projection-based reduced models). Data-fit models, which are primarily used for the evaluation of experimental data, involve the interpolation or regression of the generated results from a few runs of the high-fidelity model. PR, KG, MARS, ANN, and many other methods are examples of data-fit PMs. In data-fit models, the underlying physics does not change, and they are identified as non-physics-based approaches.

A multi-fidelity model, which is a physics-based approach, attempts to solve a low-speed, high-fidelity problem by replacing it with a high-speed, low-fidelity model [30]. Fidelity here is used to describe the level at which a model could reproduce the physics of the desired phenomenon. The process of achieving a lower-fidelity model can be fulfilled through coarser discretization [31], simplifying physics assumptions [32], etc. In an ROM, a high-fidelity model projects down into a low-dimensional system with equivalent characteristics that have fewer degrees of freedom [33]. In other words, an ROM lowers the dimensionality of the primary system by neglecting irrelevant parameters while holding the characteristics and physics over defined space. They are based on the discretization of the underlying partial differential equations of the high-fidelity models. ROMs can be grouped [20] based on the type of system (linear, partially linear or non-linear, parametric or non-parametric, and time-dependent or time-independent). Some popular techniques to solve ROMs are proper orthogonal decompositions (POD) [34], trajectory-piecewise linear (TPWL) [35], and the discrete empirical interpolation method (DEIM) [36], and they are further explained in Section 2.4.2. In some studies, the terms "multi-fidelity model" and "reduced-order model" have been interchangeably used, and they are considered within one group [21], [22]; however, the majority of the literature considers them as separate models.

There are pros and cons to each of the three types of PMs in this classification (data-fit, MFM, and ROM). Data-fit models are easy to use in low-dimensional problems; however, if the number of parameters increases, their application is problematic. Additionally, data-fit models cannot approximate functions other than first- and second-order responses, and the high-fidelity models need to be run many times [37]. To use MFMs, a significant engineering effort is required, and they are usually implemented in an opportunistic manner with the made assumptions. In contrast, the data-fit and reduced-order models are mathematically derived from the high-fidelity model [30]. Although the use of ROMs requires a good understanding of simulation codes for projection, they do not require numerous computational models (that might not always be accessible), in comparison with MFMs [15]. Data-fit models and ROMs can be compared in different aspects. One is that the procedure of generating the PMs using the data-fit models is non-intrusive, and it only needs to define the system inputs and obtain the outputs by running the high-fidelity model. On the other hand, this procedure is intrusive for ROMs, and system operators should be projected into a reduced subspace [30]. Another feature of ROMs is that they are capable of estimating the errors and bounds between the high-fidelity model and the reduced one [38]. Additionally, ROMs are considered as physics-based models; therefore, they have the advantage of better prediction accuracy compared with the data-fit models, and as they retain the underlying physics, they are even capable of extrapolating away from the initially given space. As a result, they can evolve dynamically in time [15].

Another way to classify PMs is based on time dependency, i.e., static and dynamic modeling [16]. Static PMs are typically built for one or a few discrete time steps and are not valid for processes at other times. In static modeling, only the spatial phenomenon is important. On the contrary, if temporal dependence is added, then the phenomenon is treated as dynamic rather than static, and it is called dynamic or time-dependent proxy modeling. Dynamic models are constructed for the whole desired time interval, not just discrete times [16]. There may be some spatial dependency in the context of static modeling, but no orders in time are effective. In contrast, in temporal or dynamic modeling, past, present, and future concepts exist. In a temporal definition, the sample locations happen in the past, and the predictions extrapolate the future [39]. Some of the popular techniques to deal with spatial problems are described in Section 2.3.3. To model temporal problems, other techniques, such as a recurrent neural networks, are needed. For details about the models that deal with temporal datasets, please refer to the book written by Lazzeri [40].

Mohaghegh [7] divides PMs into two main categories: traditional and smart. The previously discussed methods (data-fit, MFM, and ROM) fall into the "traditional" proxy model (TPM) category. "Smart" proxy models (SPMs) are trained using machine learning and pattern recognition techniques and require some additional steps in the development process, compared to TPMs. The development steps will be discussed in detail in Section 2.3. SPMs are capable of reproducing the high-fidelity models without reducing the physics and order of the original system, and they do not decrease the resolution of the model in time or space [41]. SPMs can be developed as grid-based or well-based models depending on the objective of the study [42]. The SPM is referred to as well-based if the objective is to make predictions for parameters at the well locations, such as production rates for oil, gas, and water [43]. In reservoir engineering, production optimization and history matching fall within this class of SPM. If it is desired to build the SPM with outputs at the grid level, it is considered grid-based. Pressure and saturation prediction for different phases are examples of parameters at the grid level, and monitoring their alteration is of paramount importance during injection scenarios where front propagation tracking is needed [44]. Gholami et al. [8] developed another class of SPM; a coupled SPM, which is a combination of well-based and grid-based models and is able to generate the results both in well and grid levels simultaneously. Gholami et al. used a workflow to transfer the parameters between grid-based and well-based models, and they used a cascading procedure to produce the results when the model moves forward in time.

If the objective for generating the proxy model is optimization, it is called surrogate model-based optimization (SBO). Bartz and Zaefferer [17] grouped the SBOs into three types: single, multi-fidelity, and ensemble models. In single SBO, there is only one model to construct the PM in the specified search space. Multi-fidelity and ensemble SBOs use more than one model to approximate the high-fidelity model; however, the multi-fidelity models describe the high-fidelity model in different levels of detail, whereas the ensemble model presents it at the same level of detail. Any of the single, multi-fidelity, or ensemble SBOs can then be coupled with either deterministic or non-deterministic optimization algorithms to optimize the output. Barton and Meckesheimer [18] described SBO in two groups based on the way they search for the optimum. One group updates in an iterative manner and looks for local optimums, and the other finds the global optimum with only one fitting chance. ANN and KG are examples of a global surrogate model, which can be coupled with a non-deterministic (evolutionary) optimization algorithm to find the global optimum. On the other hand, for the local surrogate model, we can name response surface model (RSM) as an example that can be coupled with a deterministic algorithm to find the local optimums [45].

Other classifications for PMs can be seen in the literature; for example, Jaber et al. [19] divide PMs into statistical and virtual intelligence models. The statistical models involve using RSM, and the virtual intelligence models employ machine learning techniques.

We discussed earlier in this section that Mohaghegh considered the PMs in two broad classes of SPMs and TPMs. However, PMs in this work are classified into four classes: MFMs, ROMs, TPMs, and SPMs. Each of these classes has a different development procedure. The development methodology for SPM and TPM will be discussed in Section 2.3. Figure 3 summarizes the different classes of PMs mentioned under the classification used in this review, and it groups them into four main categories.

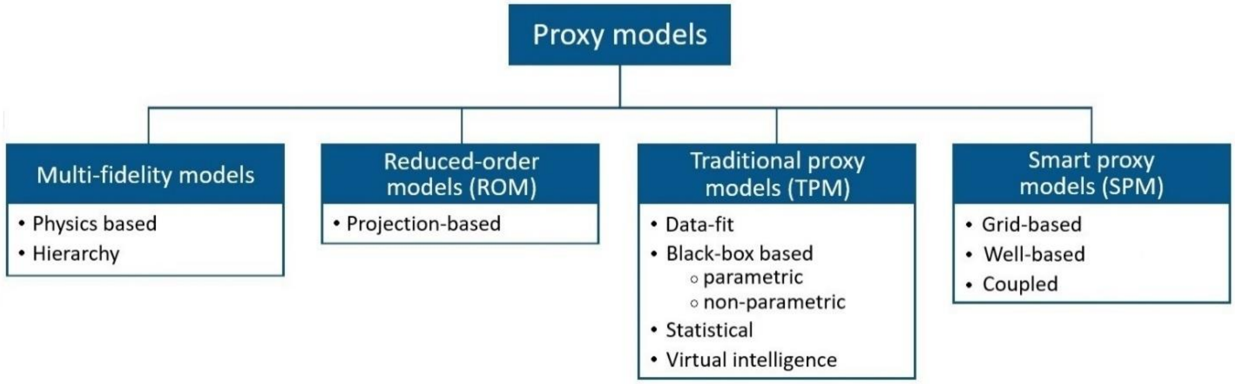


Figure 3. Newly defined proxy model categories-MFM, ROM, TPM, and SPM

2.3. Methodology

The development of ROMs is based on the projection of the problem into a lower dimensional case, and MFMs are based on simplifying physics. Detailing the procedure to develop these two proxy types is not the purpose of this work. Instead, it is more focused on the TPMs and SPMs. TPMs in this paper include all classes of PMs that have a development procedure similar to the one shown in Figure 4, such as data-fit, parametric and non-parametric, statistical or virtual intelligence, and static or dynamic. The development strategy of SPM is also presented in Figure 4 based on the strategy introduced by Mohaghegh [7].

A proxy model cannot be built without a high-fidelity model. For example, in the case of reservoir modeling, the high-fidelity model is the numerical model. The main steps to construct a TPM include the following: (1) define the objective, inputs, and output parameters for the high-fidelity model with their range, (2) perform the sensitivity analysis if needed, (3) perform the sampling and generation of different design scenarios, (4) run the high-fidelity model to produce the preliminary results, (5) train and validate a new underlying model, and (6) employ a TPM to generate results.

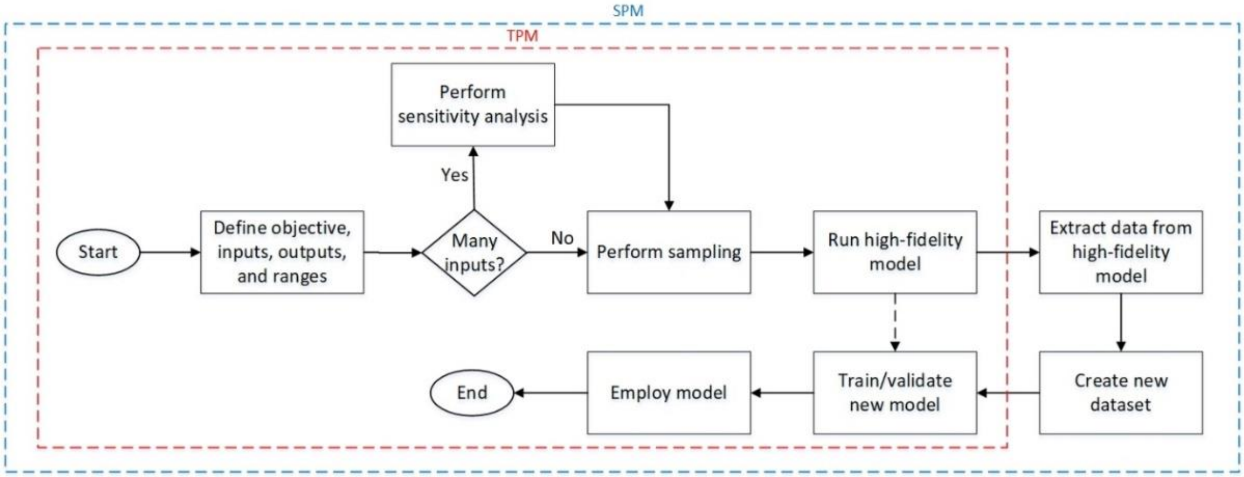


Figure 4. Traditional and smart proxy models development schematic

In the first step, which may be considered the most crucial, the reason to construct the TPM is determined. The high-fidelity model (or subsequent TPM) may be used to accomplish many objectives. Optimizing a parameter such as total production rate or watercut in a waterflooding scenario, obtaining the highest net present value, or history matching are examples of objectives. Then, depending on the objective, the essential parameters with their ranges for the output of the problem must be selected. For example, in the waterflooding case, injection rates, water viscosity, well production rates, etc., can be the effective parameters. A reservoir environment is considered a non-linear and complex problem in which many parameters play a role. Dealing with many parameters in such an environment and running the simulator based on all the parameters is sometimes very costly and time-consuming. In such a case, sensitivity analysis can find the non-influential inputs and reduce the number of them. So, sensitivity analysis is essential in constructing the PM in cases with a high number of inputs, and is performed before the sampling step [46], [47], and it can lower the dimensionality of the model [48]. This will be reviewed in Section 2.3.1. In the sampling step, we design the experiment, we distribute the parameters in their ranges, and the high-fidelity model runs at these designed selections of parameters. Section 2.3.2 discusses the sampling techniques in more detail. After running the high-fidelity model and recording the desired output, the new machine learning model trains and validates the inputs and outputs. An issue in this step is the significant number of potential modeling techniques, and sometimes it is difficult to determine what technique is suitable for specific applications. A brief description of the different models is presented in Section 2.3.3. Finally, the built model, which is referred to as the TPM, can predict the output parameter in a much shorter time for the initially selected design parameters. The TPM can then be used for other purposes such as SA, optimization, uncertainty quantification, etc.

The creation of an SPM has a few additional steps compared to the TPM, as shown in Figure 4. The most crucial part of the SPM, like TPMs, is determining the objective for model construction. Then, a set of effective parameters is picked, and sampling is performed within the specified range for the parameters. In the next step, the high-fidelity model is run to generate different realizations of the model. Then, a combination of static and dynamic parameters (including both inputs and outputs) is extracted from the high-fidelity model, and they are used to create a new database [7]. Generating new parameters is called feature engineering, which leads to finding new hidden patterns in the dataset. In TPM construction, we do not involve this step, and the proxy only makes a relationship between the initial design parameters and the output. Hence, SPM provides higher accuracy in approximating the output parameters compared to TPM. Feature engineering also helps SPM to accurately predict the grids' characteristics. In comparison, TPM suffers in grid level predictions because a trained model on a limited number of input parameters cannot predict the complex changes in all grids. Neither a TPM nor SPM solves the mathematical equations of a high-fidelity model. They basically decouple the equations, constraints, and complexities of the problem into a numeric dataset. This dataset then can be used to train a(n) AI/statistical model to approximate the desired output. Decoupling the equations makes the TPM and SPM superior to MFM or ROM. It is worth mentioning that the MFM and ROM are the simplified or reduced form of the high-fidelity model, and their construction still needs a great knowledge of the problem. However, TPM and SPM require less scientific knowledge, and they need less time and effort in development.

For a well-based dataset, the rows (observations) in the new dataset are the wells in the reservoir model, and the columns are the extracted parameters from the high-fidelity model at the well level and various time steps. However, for a grid-based dataset, the rows are the individual grids within the high-fidelity model, and the columns are parameters extracted at the grid level and various time steps. For example, Gholami et al. [8] considered various parameters such as porosity, permeability, pressure, saturation for phases, location of the grids, distances to the boundary and closest offset well, production data, bottom-hole pressure (BHP), etc., as the parameters of the grid-based dataset. The authors introduced a tiering system to consider the impact of the surrounding grids as well, and the static and dynamic parameters related to the tiers were imported into the dataset. Such datasets for the generation of the smart proxies would be massive, and reducing both the number of observations and parameters prior to the proxy development would be needed. In the aforementioned work [8], the authors reduced the size of the dataset (from 396,000 to 55,000 observations, and over 1000 parameters to 310 only for one of the realizations out of 13) through grid lumping in the Z-direction and feature selection techniques. The number of parameters in the dataset depends on the number of introduced tiers, offset wells, and the previous timesteps that are going to be cascaded into the new timestep. After forming the dataset, the rest of the steps are similar to those of the TPM.

The main advantage of using an SPM is that it only takes a few seconds to approximate the full reservoir. However, this is only an advantage if the SPM construction time is reasonable. An SPM produces a massive amount of data, which are extracted from different scenarios or realizations of the high-fidelity model. So, picking a small number of sample points is required to avoid generating such a massive dataset and to decrease the construction time [49]. The number of required runs usually depends on the geological properties and operational constraints, and it should cover as much necessary information as possible, depending on the purpose of the study. For instance, in the work carried out by Gholami et al. [8], the total number of realization runs was only 13 for the four design parameters, or in the research by He [9], the number of runs was only three for the two parameters of porosity and permeability in a history matching process. SPM is a novel approach and expects to provide higher accuracy than TPM. TPM approximates the outputs only based on the initial designing parameters, while SPM creates an approximation by involving many new parameters. The newly introduced parameters usually have positive importance on the outputs [7]. The usage of SPM is more significant in grid-based models where it can predict the outputs at the grid level. A TPM with only a few input parameters lacks the prediction ability for the individual grids in the reservoir. There are only a few SPM examples in the literature, and they almost used the same strategy for development [8]–[13]. We suggest that conducting a sensitivity analysis comparing various techniques for the main steps involved in SPM construction can improve the overall SPM accuracy and development time. For example, only the ANN model for the underlying model construction step, stationary Latin hypercube sampling for the sampling step, or similar tiering systems have been tested for the data extraction step. As a result, it is crucial to know the main techniques for each step. Then, it is important to apply the techniques for future research, and compare them to check how each technique improves the SPM in terms of accuracy and construction time. The next sections provide a comprehensive review of the existing procedure and specific step techniques required in TPM and SPM.

2.3.1. Sensitivity Analysis

Sensitivity Analysis can be used to study how the outputs of a model change due to the variation in the inputs of the specified ranges. The input parameter is called “sensitive” if the variation in its range significantly changes the output. Additionally, a parameter is considered “insensitive” or “robust” if the output does not change a lot [50]. Sensitivity analysis establishes the importance of parameters and the inner workings of the models, which can lower the dimensionality of the model [48].

Performing a sensitivity analysis can be significant; for instance, a reservoir model that is representative of a very non-linear environment includes many parameters. As a result, many sample points are required to cover all the search space for the parameters, and running a high-fidelity model for all the designed points is very costly. In such a case, with a large number of parameters for the model, sensitivity analysis is an essential step in generating the TPM, and is performed before the sampling step [46], [47]. Consequently, sensitivity analysis can find the non-influential parameters and reduce the number of inputs. Sensitivity analysis can sometimes be considered as an objective for constructing a TPM [48], [51]. The relationship of inputs with the output in a model could be complex and challenging. In this case, a TPM can be constructed based on all the inputs. This TPM can then be used for sensitivity analysis and run numerous times to find the effect of inputs on the output very quickly.

sensitivity analysis is categorized into local sensitivity analysis (LSA) and global sensitivity analysis (GSA) [52]. In LSA, the inputs are subjected to small perturbations at specific points, and the changes in the output parameter are studied. LSA is also known as the one-at-a-time or univariate method [52], [53]. It studies the derivative of output to only one input parameter while keeping the rest of the inputs constant. It cannot investigate the effect of all inputs varying at the same time. LSA is popular for models with a low uncertainty level [54], but it is not suitable for complex reservoir modeling problems. On the other hand, in GSA, the behavior of the output over the entire range of the inputs is studied [52]. Some of the common methods for GSA are based on Monte Carlo (MC) sampling, Sobol, or Morris methods, which are based on probability distributions [55]. MC-based sampling methods allow us to analyze the influence of the parameters, but using them is computationally expensive [56]. The MC-based sampling methods can also be used to find the global optimum of a problem. The Sobol method works based on the variance decomposition theory. It investigates the interaction and contribution of input parameters to the output in a certain number of sample points compared to MC [57]. The Morris method, an extension of LSA, studies multiple points in the parameter range instead of only one [58]. More information on GSA methods can be found in Song et al. [59]. Typically, GSA methods that are based on probability distributions need more runs of the high-fidelity model. As a result, they are computationally more expensive compared to LSA.

2.3.2. Sampling

Sampling is defined as the process of obtaining data points over the search space of the parameters to be able to construct a PM. The quality and performance of a PM depend strongly on the number of sample points in the specified range of parameters [60]. As discussed earlier, one of the merits of using PMs is to provide a fast and accurate duplicate of the high-fidelity model. Regardless of the underlying model, having a large number of sample points to construct the PMs eventually results in an accurate model; however, the computational cost in the generation of the PMs would be significant. Another effective factor in increasing this cost is the number of implemented parameters in the process of creating the model, which has a direct relationship with the number of sample points as well. High computational cost in the construction of the PMs due to high number of parameters diminishes their applicability compared with the high-fidelity model. This is known as the curse of dimensionality [14], [61]. This cost is usually incurred prior to the implementation of the PM in running the high-fidelity model at sample points and steps in designing the model. The objective in sampling is to acquire the maximum information using the minimum possible number of sample points or the minimum number of high-fidelity model runs. Therefore, the selection of a proper sampling strategy to construct a trustworthy PM is of immense importance. Consequently, once the PM is constructed, hundreds or thousands of runs can be completed in a fraction of a second, which is essential in production optimization and field development planning in the oil and gas field.

Sampling strategies are grouped into the two broad classes of stationary (also known as one-shot, static, priori, domain-based, or model-free strategies) and sequential (also known as model-based, adaptative, or posteriori) sampling. The workflow of both techniques is demonstrated in Figure 5.

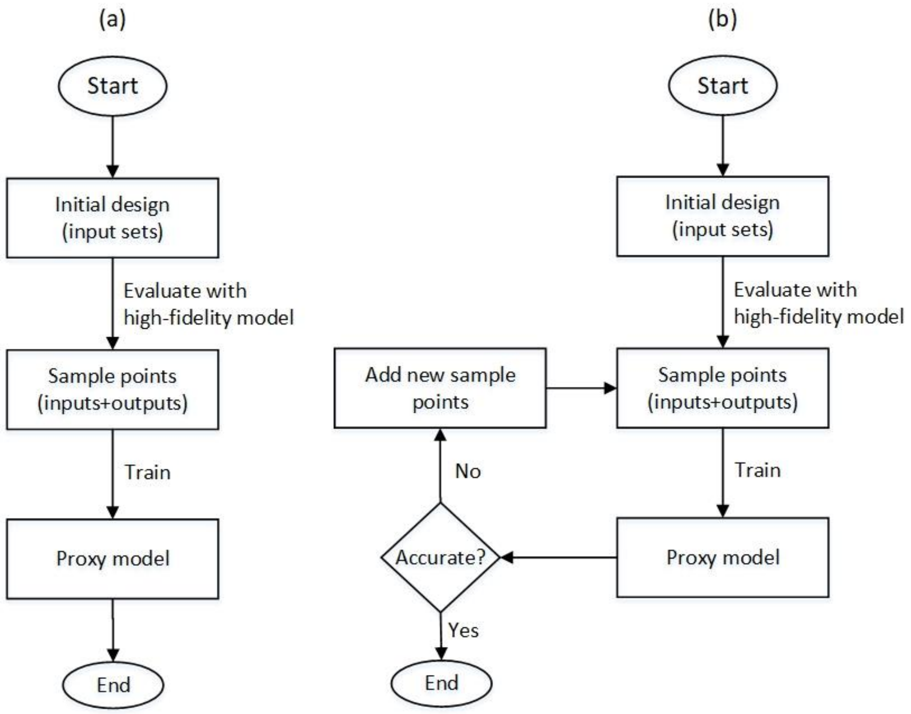

(a)
Start
Initial design (input sets)
Evaluate with high-fidelity model
Sample points (inputs+outputs)
Train
Proxy model
End
(b)
Start
Initial design (input sets)
Evaluate with high-fidelity model
Add new sample points
Sample points (inputs+outputs)
No
Train
Accurate?
Proxy model
Yes
End

Figure 5. Workflow for (a) stationary and (b) sequential sampling (Modified from Ref [62] published by Ghent University, 2011)

In stationary sampling, the sample points are distributed throughout the whole design space based on a pattern [63]. In stationary sampling, after the selection of the design points, the high-fidelity model is evaluated at each of the specified input sets, and the corresponding output is obtained. Consequently, the PM is constructed, and it can be applied for further decisions. The advantages of stationary sampling are ease of deployment and uniform coverage of the domain. However, if the performance of the PM was not approved at any stage, the whole procedure should be performed from the beginning with new and more numerous sample points, which increases the cost of computation. The situation becomes even worse when the problem is highly non-linear and many parameters are involved in the process of constructing the PM, which together increases the number of sample points and makes the modeling inefficient and unreasonable [64]. The stationary sampling methods focus on a uniform fill-up of the domain with sample points, and they usually have a fixed pattern in all of the cases under study. Some famous techniques in this category are factorial designs (full and factorial designs), optimal designs, Latin hypercube sampling (LHS), orthogonal array sampling (OAS), and random sampling such as MC.

In the simplest case of full factorial design, which only considers two levels for each parameter, 2^K sample points are generated, where k is the number of parameters. The sample points are maximized in the distance using the full factorial design; however, the number of sample points increases rapidly when the number of parameters increases [65]. The fractional factorial design, a sub-class of the full factorial design, does not consider some sample points in the full factorial design without losing information, making it viable for higher dimensions [66]. Different designs, such as the Plackett–Burman design (PBD), central composite design (CCD), Box–Behnken design (BBD), and Taguchi, fall into the category of fractional factorial designs [67]. In optimal designs, the parameters are determined without bias and with minimum variance; hence, they require a lower number of sample points [68].

LHS, as a stratified sampling technique, divides each parameter into N bins or equal intervals. Then, each bin for each of the parameters fills with only one sample point. In LHS, there is control over the number of sample points, which is a big advantage in the construction of a PM, where we are looking to limit the number of high-fidelity runs [69]. However, not all Latin hypercube designs uniformly distribute the sample points in the domain, and it is necessary to optimize the space-filling procedure. Viana et al. reviewed some of the techniques to optimize the LHS to better fill up the domain under study [70].

OAS is a generalization of LHS, and it uniformly distributes the sample points in the dimensional projection of the parameter dimensional domain [62], [71]. In OAS, four parameters—sample point size, domain dimension, the number of bins per dimension, and the strength—need to be defined. For example, OAS (9, 2, 3, 2) includes nine sample points, two domain dimensions, three bins per dimension, and a strength of two. LHS is considered as an OAS with a strength of one. To understand the construction procedure, and for details about OAS, refer to the work carried out by Hedayat et al. [71]. The selection of the four parameters of OAS is sometimes challenging. Additionally, different selections for bins and the placement of samples in the bins might exist for each problem, limiting OAS usage [64]. In MC, the independent samples are randomly generated, and this process is repeated many times to achieve the desired quantity.

In sequential sampling, the sample size starts with an initial and limited number of points, and new data points are added to the existing ones. The model then stops once it reaches a preferred performance and accuracy [72]. Hence, using this approach, the total time to develop a PM is significantly reduced, as less deployment of the high-fidelity model is needed than when using the stationary sampling technique [73]. There are two main objectives in sequential sampling, known as exploration and exploitation. Exploration means evenly filling up the search space and avoiding redundant new samples. The intention of the exploration part of sequential sampling is the same as stationary sampling, and it tries to find new pivotal regions such as discontinuities and the global optimum. In exploration, the new samples are chosen only based on the initial samples, not the responses of the high-fidelity model [62]. The methods with exploration objectives are usually referred to as exploration-based sequential sampling. Markov chain Monte Carlo (MCMC), low-discrepancy sequence methods (such as Sobol and Halton), nested LHS, and quasi LHS are examples of this subcategory.

MCMC is based on a sequence of random sampling from a probability distribution. In MCMC, the sample points are generated systematically in such a way that the new sample point is probabilistically dependent on the prior sample point [74]. In the low-discrepancy sequences, the term “low-discrepancy” implies that the sample points are, more or less, equally spread within the domain of parameters. A low-discrepancy sequence is a sequence with the property that all the new points are as far as possible from the existing points. In Sobol and Halton sequences [75], [76], the sample points are randomly produced within the parameters’ domain without overlapping the existing points in a progressive manner. However, Sobol outperforms Halton as the number of dimensions increases [77]. We can use nested and quasi-LHS to sequentially generate the sample points for LHS. Nested LHS includes various designs requiring one as the subset of another one [78]. The evaluation performs on the smallest subset, and if the results are not desirable, the superset could be evaluated. This process can continue to the next supersets. Here, each subset is known as a layer, which is an LHS with a level of accuracy [79]. In quasi-LHS, the new sample points are added to keep the distance from the existing design points [80].

Exploitation-based sampling techniques look for regions that have already been recognized as key domains and add the new sample points at each iteration focused on these regions. In exploitation, unlike exploration, the new samples are added only based on the information provided by the high-fidelity model. The LOLA-Voronoi sampling method [62] falls into this subcategory. This method works based on the local gradient approximation to find the non-linearity of the problem under study, and it picks more sample points near the non-linear region to increase the chance to find the local optimum.

2.3.3. Popular Models for PM Construction

PMs, as discussed in Section 2,2, can fall into different categories. This section discusses the most prominent underlying models that can be used in the construction of TPMs or SPMs.

2.3.3.1. Polynomial Regression (PR)

PR is the most straightforward technique to construct a PM. It fits a non-linear relationship at any order between the inputs (x_i) and the output (y). PR is also known for the response surface method. It is usually applicable for lower-dimensional problems and is not suitable for high-dimensional and highly non-linear systems. PR can be given by the general form of Equation (1).

$$\hat{y}(x_1, x_1, \dots, x_m) + \sum_{s_1=1}^{m} \beta_{s_1}(x_{s_1}) + \sum_{s_1=1}^{m} \sum_{s_2=s_1}^{m} \beta_{s_1 s_2}(x_{s_1} x_{s_2}) + \cdots$$
$$+ \sum_{s_1=1}^{m} \sum_{s_2=s_1}^{m} \cdots \sum_{s_p=s_{p-1}}^{m} \beta_{s_1 s_2 \dots s_p}(x_{s_1} x_{s_2} \dots) \tag{1}$$

where ŷ is the approximated output, m is the number of dimensions or parameters, β is the polynomial coefficient, and p is the polynomial order. To construct a high-order polynomial, a large number of training points are needed. Furthermore, high-order PR models cause instability and sometimes yield a false optimum [81].

2.3.3.2. Kriging (KG)

KG or Gaussian process regression is a technique to interpolate the output based on the Gaussian process by considering the prior covariances [82]. In 1989, Sacks et al. [83] used the KG to construct a PM for the first time in engineering. There is always a residual error ε between a PM response (ŷ) and the high-fidelity model response (y).

$$y(x) = \hat{y}(x) + \varepsilon \tag{2}$$

The basic assumption behind most PMs is that this residual error between the responses of models is independent. However, in KG models, this error is dependent on another term [84]. KG involves using a polynomial function f(x) and a random function (i.e., stochastic process) Z(x).

$$y(x) = f(x) + Z(x) \tag{3}$$

The polynomial term determines the global trend of the data, and the stochastic term accounts for the deviation of the output from the polynomial term. The stochastic term $Z(x)$ is assumed to have zero mean and variance of σ^2. In many problems, the polynomial term $f(x)$ can be replaced by a constant without losing performance, and this is known as ordinary KG [85].

A KG proxy model is suitable for low-order non-linear and large-scale problems, and it can work for a wide range of sample sizes and designs. However, applying it to a large-scale problem might be time-consuming [86]. It also does not work accurately for problems containing discontinued parameters and a dimensionality higher than 20 [87].

2.3.3.3. Multivariate Adaptive Regression Splines (MARS)

MARS, introduced by Friedman [88], is a regression algorithm that implements linear regression modeling for the sub-intervals of each design parameter. The location at which the sub-intervals connect to each other is known as a knot. MARS is an extension of linear regression models, and it uses a set of coefficients and basis functions to make a relationship between inputs and the output. The process of constructing MARS happens in forward/backward iterations for different sub-intervals. First, it creates a basis function (i.e., spline) for each sub-interval with its corresponding linear regression and coefficients. In the forward procedure, it looks for the optimum locations to place the knots.

$$\hat{y}(x, \beta) = \sum_{m=1}^{M} \beta_m B_m(x) \tag{4}$$

where β_m is a vector of regression coefficients, M is the number of basis functions, and B_m is the basis function which can be described as

$$\mathrm{B_m(x)} = \prod_{\mathrm{i=1}}^{\mathrm{k_m}} \mathrm{b_{i,m}\,(x_{v(i,m)} - t_{i,m})]_+^q} \tag{5}$$

where k_m is the number of design parameters in the m^th^ basis function, $b_{i,m} = \pm 1$, v(i,m) is for labeling the design parameters, $x_{v(i,m)}$ is the i^th^ parameter in the total of k parameters, $t_{i,m}$ is the knot location of the corresponding parameter, q is the order of the spline, and + subscript is for the positive part of the function (it is zero for negative values of the function).

$$\left(\mathrm{x_{v(i,m)} - t_{i,m}}\right)]_+^{\mathrm{q}} = \begin{cases} \left(\mathrm{x_{v(i,m)} - t_{i,m}}\right)]^{\mathrm{q}} & \text{if} \quad \left(\mathrm{x_{v(i,m)} - t_{i,m}}\right) > 0 \\ 0 & \text{if} \quad \left(\mathrm{x_{v(i,m)} - t_{i,m}}\right) \leq 0 \end{cases} \tag{6}$$

In the forward step, the algorithm creates functions and looks for the locations of the knots to improve the performance, which might cause overfitting. In the backward step, MARS prunes the non-influential design parameters based on the generalized cross-validation techniques [89].

MARS is suitable to deal with large and high-dimensional datasets. It is capable of performing feature selection through the backward step; however, choosing the knot locations is challenging, and it sometimes faces overfitting [90].

2.3.3.4. Artificial Neural Networks (ANN)

ANN works based on the biological neural system in the brain, which consists of many chemically connected neurons [91]. Neurons are placed in three main layers: input, hidden, and output layers. Neurons receive signals from neurons in the previous layer and transmit them to the next layer. Each neuron is accompanied by a weighting factor that should be adjusted. Adjusting the weights usually happens via an optimization algorithm, most often via the backpropagation technique [87]. The receiving signal for a neuron in the hidden layer multiplies into the adjusted weights and sum. Then, an activation function is applied to the summation to generate the output for that neuron. Some commonly used activation functions are linear, sigmoid, rectified linear unit (ReLU), and hyperbolic tangent [92]. There are some controlling parameters that should be optimized to have an accurate ANN model, such as the number of hidden layers, the number of neurons in each layer, and the activation function. ANN can approximate the problems with unknown nature, but sometimes finding the optimal controlling parameters is challenging. Additionally, using the ANN is computationally expensive, and it requires high memory. Recurrent neural networks (RNN) and convolutional neural networks (CNN) are two well-known and robust types of ANN. RNN works similarly to backpropagation ANN, with the difference that RNN has a memory to store information, and is suitable for a sequence of data and timeseries data [93], whereas CNN adaptively learns spatial patterns within the parameters, and is designed to process data in grid formats such as images [94].

2.3.3.5. Radial Basis Function (RBF)

RBF is a network that includes an input layer of nodes, a hidden layer with a radial basis function (kernel), and an output layer of linear weights [95]. This network approximates the problem in a feed-forward process through Equation (7).

$$\hat{y}(x_p) = \sum_{i=1}^{N} w_i \phi_i(x_i, x_p) \tag{7}$$

where x_p is the vector of inputs, x_i is the ith center of the total N radial functions, $\phi_i(x_i, x_p)$ is the ith kernel to calculate the distance between x_i and x_p, and w_i is the corresponding weight factor for the radial function. The radial function can be selected in different forms such as linear, cubic, thin plate spline, Gaussian, multi-quadratic, and inverse multi-quadratic. Several methods exist to define the centers of the radial functions, such as the orthogonal least-squares method [96].

RBF, like ANN, is easy to implement; however, RBF is suitable for training noisy datasets because of its non-linear characteristics. RBF is not recommended for problems with a high number of parameters that would be very expensive to compute.

2.3.3.6. Support Vector Regression (SVR)

SVR was first developed by Vapnik [97] to find a function to relate the inputs and the output. It can approximate the problem based on the weighted sum of basis functions added to a constant term as

$$\hat{y}(x) = \mu + \sum_{i=1}^{n} w_i \phi_i(x_i, x) \tag{8}$$

where μ is a constant bias term, ϕ is the basis function, and w_i is the corresponding weight factor for the basis function. For the basis function, either linear or non-linear functions can be considered. The main idea behind the SVM is to look for the best fit line (hyperplane) and the boundary lines to obtain the maximum number of data points. The formula used to approximate the output in SVR is similar to RBF; however, the methods they use to obtain the unknowns are different. In RBF, the unknowns are determined by minimizing the error between the actual and predicted outputs, but in SVR, μ and w are obtained through solving an optimization problem for the threshold between the hyperplane and the boundary lines [98].

SVR has the advantage of accurate and fast prediction, and it is robust to outliers. Furthermore, SVR is suitable for high-dimensional problems with non-linear data. However, training SVM and selecting the appropriate parameters is challenging [99].

2.3.3.7. Genetic Programming (GP)

GP was initially proposed by Koza [100], which is an extension of the genetic algorithm (GA), and it is used mostly for symbolic regression. Both techniques are based on evolutionary Darwinian theory. They start with a population of random solutions (known as parents), and the solutions evolve through generations by dropping the not-fitted solutions. In GA, these candidates are in the form of coded strings (also known as chromosomes), while they are in the form of mathematical expressions in GP [101]. GP uses a tree structure to represent the mathematical expressions. The two main components of these trees are the function set (nodes) and terminal set (leaves) [100]. The function set can be chosen through mathematical operators, functions, or conditional statements, and the terminal set includes constants and problem parameters. In GP, the evolution mainly happens in two different processes, mutation and crossover. Mutation involves substituting a random new segment with a segment in the parent, and crossover happens by exchanging segments between two parents and forming two new solutions (offspring). The algorithm also reproduces random solutions to compensate for the dropped solutions in the previous generation. All the solutions again check for the fitness in the new generation, and this process continues until an acceptable fitness value is achieved or the algorithm reaches its generation limit.

One of the drawbacks of using GP is that it needs to be run multiple times because of the stochastic nature of the algorithm. The fixed value of the algorithm parameters (such as crossover and mutation probability) constructs different models at each run. Too-simple models may result in poor predictions, and too-complex models may cause overfitting [102]. On the other hand, GP can generate a high number of potential solutions without considering the underlying assumptions of the problems.

2.3.3.8. Random Forest (RF)

RF is an ensemble learning technique initially developed by Breiman [103]. RF is an ensemble of multiple unpruned decision trees. It implements the bootstrap sampling (bagging) technique, which uses a random selection of the dataset as the training set and uses the rest of the dataset as the testing set for each tree in the ensemble. The random selection in RF helps the diversity in the ensemble of trees and improves the predictions, and prevents overfitting [104]. RF gives the final result as an average of the results from individual trees, also known as aggregation.

In the process of growing individual trees, a small subset of size m out of M parameters for each node (m < M) will randomly be picked to train the trees. The size of m is kept constant during the forest growing, but the parameters are changed for each node. This process helps the model to use all the potential parameters for the prediction and prevents the model from relying on any specific parameters. Consequently, the best split at each node is chosen among m parameters rather than all the parameters of M. Hence, RF involves using the feature selection technique. It analyses all the parameters without deleting them and selects the influential ones [105]. As a result, RF is a suitable algorithm for large datasets with a high number of parameters, and it produces very accurate results even when a part of the data is missed. However, it lacks prediction ability beyond the training data range [2].

2.3.3.9. Extreme Gradient Boosting (XGboost)

XGboost is a supervised learning method proposed by Chen and Guestrin in 2016 [106], which is based on the gradient boosting machine (GBM) technique introduced by Friedman [107]. In the GBM technique, the algorithm sequentially adds the input parameters to an ensemble of decision trees to help to improve the prediction. Unlike the RF, which is an ensemble of deep independent decision trees, GBM is an ensemble of shallow trees. GBM builds only one decision tree at a time, and it sequentially improves the ensemble's performance as it goes forward to the next tree [108]. RF involves averaging the results from all the independent decision trees, while GBM calculates the loss function for each tree. XGboost differs from GBM in the way it minimizes the loss function. XGboost uses the second-order gradient of the loss function, which helps to more easily minimize the function. Additionally, the parallel computing ability and the implementation of some generalization terms to prevent overfitting are other benefits to using the XGboost compared to the GBM [109]. Some of the main disadvantages of XGboost are the high training time for large datasets and the inability to predict beyond the training data range.

2.3.3.10. Polynomial Chaos Expansion (PCE)

PCE was introduced first by Wiener [110] to project the output on an orthogonal stochastic polynomial basis function in the random inputs. The general form of the PCE can be defined as Equation (9) [111]:

$$\hat{y}(X) = \sum_{\alpha \in \mathbb{N}^M} \beta_\alpha \Psi_\alpha(X) \tag{9}$$

where α is the index with M dimensions, β_α are the deterministic polynomial chaos coefficients, $\Psi_\alpha = \{\Psi_1, \Psi_2, \ldots, \Psi_M\}$ is a set of multivariate orthogonal polynomial basis, and X is the vector of input parameters with M dimension. The multivariate orthogonal polynomial basis can be written as the product of univariate polynomials $\phi_{\alpha_k}$of degree α_k:

$$\Psi_\alpha(X) = \prod_{k=1}^{M} \phi_{\alpha_k}(X) \tag{10}$$

where ϕ_{α_k} is the univariate orthogonal polynomial in the kth parameter of degree α_k. There are different univariate polynomial families such as Hermite (based on Gaussian distribution), Laguerre (based on gamma distribution), and Jacobi (based on beta distribution) [112].

The main benefit of using PCE is that as the order of expansion increases, it guarantees the convergence of ŷ to y, preventing the drawback of overfitting in many other techniques. Additionally, PCE is almost applicable to all input distribution types [113]. However, PCE sometimes slowly converges, and if the order of expansion increases in high-dimensional problems, it requires a large number of high-fidelity model runs. A summary of the aforementioned modeling techniques with their pros and cons is presented in Table 1.

Table 1. Advantages and disadvantages of different modeling techniques.

Modeling Technique	**Advantages**	**Disadvantages**
PR	• Suitable for lower-dimensional problems.	• Not suitable for high-dimensional and highly non-linear systems. • A large number of training points is needed to construct a high-order polynomial [81]. • High-order PR models cause instability and sometimes yield a false optimum [81].
KG	• Suitable for low-order non-linear and large-scale problems. • Works for a wide range of sample sizes and designs.	• Applying it to a large-scale problem might be time-consuming [86]. • Not accurate for the problems containing discontinued parameters and the dimensionality higher than 20 [87].
MARS	• Suitable to deal with large and high-dimensional datasets • Capable of doing the feature selection through the backward step	• Choosing the knot locations is challenging, and it sometimes faces overfitting [90].
ANN	• It can approximate the problems with unknown nature • Easy implementation	• Finding the optimal controlling parameters is challenging. • It is computationally expensive, and it requires high memory.
RBF	• Suitable for noisy datasets • Easy implementation	• Very expensive computation for problems with a high number of parameters.
SVR	• It is accurate with fast prediction • It is robust to outliers. • Suitable for high-dimensional problems with non-linear data.	• Training and selecting the appropriate parameters is challenging [99].
GP	• Can generate a high number of potential solutions without considering the underlying assumptions of the problems.	• Needs to be run multiple times because of the stochastic nature of the algorithm. • Too-simple models may result in poor predictions. Too-complex models may cause overfitting [102].
RF	• Suitable for large datasets with a high number of parameters. • It produces very accurate results even when a part of the data is missed.	• Lacks the prediction ability beyond the training data range [2]
XGBoost	• Able to do parallel computing • Implementation of some generalization terms to prevent overfitting	• High training time for large datasets • Unable to predict beyond the training data range.

PCE	• Almost applicable to all input distribution types • As the order of expansion increases, it guarantees the convergence of prediction, and prevents overfitting	• It slowly converges. • If the order of expansion increases in high-dimensional problems, it requires a large number of high-fidelity model runs.

2.3.4. Optimization

Once the PM with the chosen underlying model is built and evaluated for its accuracy and fitness, the model is ready for further use, such as in predictions and optimization. In the optimization process, to find the best selection of the design parameters, an objective function based on the purpose of the problem should be defined.

Optimizers are divided into deterministic (gradient-based) and stochastic (meta-heuristic) approaches. Deterministic methods implement the gradient of the objective function, and they are suitable for objective functions with a smooth surface. The biggest drawback of deterministic approaches is that their performance depends on the initial guess of the design parameters, and they may become trapped in a local optimum, preventing them from finding the global optimum. Deterministic methods can be categorized into ensemble-based, simultaneous perturbation, and adjoint methods [114].

Stochastic approaches solve the problem by borrowing rules from nature. Popular methods in this category are GA, particle swarm optimization (PSO), simulating annealing, ant colony optimization (ACO), and differential evolution (DE), which are the standard methods to find the global optimum.

GA is an evolutionary algorithm proposed by Holland [115], and it uses Darwin’s rule and creates solutions by implementing processes such as mutation and crossover, which was previously discussed in Section 2.3.3.7. GA is a strong method to find the global optimum that works for both continuous and discrete optimization problems. GA can be parallelized and is applicable to multi-objective functions in global optimization problems; however, it is computationally expensive and time-consuming [116].

PSO, like GA, is a population-based method, and it uses the collective behavior of animal groups such as birds and fish. PSO was first introduced by Kennedy and Eberhart [117]. The algorithm generates random solutions in which each solution is known as a particle, and the group of particles forms a swarm. Each particle in the swarm moves in the searching space of the design parameters. The position and velocity of the particles continue to update at each step based on the individual and global best solutions at the previous step until the particles find the global optimum. PSO has less computational burden but is more reliable in finding the global optimum compared to GA. Furthermore, the PSO approach is less effective for problems of more than three input parameters [116].

Simulating annealing is a probabilistic method that is inspired by the annealing process in metals [118]. Simulating annealing allows finding the global optimum in a large search space with the ability to jump out of any local optimum it finds [119]. In the simulating annealing algorithm, the controlling parameter is called temperature. The algorithm starts with a positive temperature, and this temperature decreases gradually to a zero value. At each step, a random solution close to the previous solution is generated, and it tries to move the temperature-dependent probability toward zero. In other words, while the algorithm searches the working space, the probability of accepting worse solutions gradually decreases. Simulating annealing is suitable for problems that contain many local optimums, and it also works on problems with discrete search space [120].

ACO is also a population-based algorithm and works based on the behavior of real ants when they are searching for food [121]. Real ants find the shortest path between a food source and the colony by communicating with each other and following substances named pheromones. If they sense a pheromone in their vicinity, they reinforce their movement toward that path. Similarly, in optimization problems, artificial ants iteratively search the domain space of parameters for the best solution in different generations. When an ant finds a solution, it marks the path to that solution as a transition rule and deposits an amount of pheromone on that route. The fitness of that path or the solution is determined based on the amount of pheromone left on that path. In the next generations, the ants are guided by the pheromone concentration left by the previous generations toward better solutions. ACO shows similar advantages and disadvantages to PSO, and it is not suitable for problems with more than three dimensions [116].

DE, as a population-based algorithm, was first proposed by Price et al. [122] for global optimization over continuous search space. Despite its name, this method does not require any calculations for the gradient, and the problem does not need to be differentiable. DE is a robust technique that converges to the solution, requiring only a few controlling parameters. The algorithm starts searching the design space by creating a population of random solutions and forms new solutions by combining the existing ones. DE, like GA, involves using mutation, crossover, and selection, which help the solutions evolve at each generation. However, DE deals differently with mutation and crossover. DE performs the mutation by creating a mutant vector of three randomly selected vectors and performs the crossover by creating a trial vector of the mutant vector and target vector [123]. Then, the fitness of the trial and target vectors is evaluated, and the best is kept for the next generation [124]. In DE, the selection of the parent solution is not based on fitness. In contrast, every solution is selected as a target vector (one of the parents); therefore, all the vectors have the chance to be one of the parents. DE only needs to adjust three parameters, and it has lower computational complexity compared to GA.

Additionally, the algorithm stops the solution from being trapped in local optimums [125]. Nevertheless, adjusting the controlling parameters is sometimes challenging [126].

2.4. Application of Proxy Models in the Oil and Gas Industry

This section discusses the application of different proxy types in the oil and gas industry, especially in reservoir modeling. Based on the literature, data-fit models are prevalent, and the application of MFMs and ROMs is limited. This review tries to cover the comprehensive implementation of all types of proxy models in different areas of the subsurface environment. Various applications of proxy modeling such as SA, uncertainty quantification, risk analysis, history matching, field development planning, and reservoir characterization are presented in this section. Furthermore, implemented cases for other models are briefly discussed in this section.

2.4.1. Multi-Fidelity Models (MFM)

MFMs try to reduce the physics of the problem. Streamline modeling, upscaling, and capacitance-resistance modeling (CRM) are the most popular techniques of MFMs in reservoir modeling. Streamline models decouple the governing flow equations in a reservoir along one-dimensional streamlines, and as a result, they boost the speed of calculation [127]. Streamline modeling has been applied in a variety of subsurface problems, such as production optimization, mainly through waterflooding [128], [129], uncertainty quantification [130], history matching [131]–[134], and well placement optimization [135]. Streamline models are often applied to a fine-scale reservoir model [136], and they need to run the high-fidelity model at different time steps. Consequently, the speedup capability of streamline models is limited [137]. In upscaling, as another way to simplify the physics, the equivalent petrophysical properties at a coarser scale are calculated [138]. Upscaling has been implemented for a wide range of objectives in reservoir modeling [139]–[142].

The idea of duplicating the subsurface behavior using a circuit of capacitors and resistors was first presented by Bruce in 1943 [143]. He used this concept to mimic the behavior of a strong water drive reservoir. This was achieved by comparing the governing equations of electrical circuits and porous media; the potential difference is the motive for the electrons to flow in electrical circuits while the pressure difference is the main reason for the fluid flow in porous media. Both systems have the characteristic of storing energy. In subsurface porous media, compressibility causes the fluid to accumulate, but the electrons are stored in capacitors. CRM was first presented by Yousef et al. [144]. The proposed model was capable of mimicking the porous media behavior between injectors and producers to identify the transmissibility trends and flow barriers. CRM estimates the values for parameters by relating the input and output signals. It considers the pressure changes caused by injectors and the aquifer as the inputs, production rates as the outputs, and the properties of rock and fluid (such as compressibility and saturation) as the related parameters. The CRMs can provide an insight into the inter-well connectivity, drainage volume, and reservoir heterogeneity, for example, by channeling along the layers [145]. Furthermore, they can be applied for history matching and production forecasting, requiring only production/injection rates and BHPs [146], [147].

In general, any technique that tries to solve the problem by simplifying the underlying physics is an example of an MFM. For example, in work carried out by Wilson and Durlofsky [148], in which a dual-porosity, dual-permeability reservoir model was simplified into a single-porosity, single-permeability model, the model can be considered as a reduced physics or a MFM proxy approach.

2.4.2. Reduced-Order Models (ROM)

The popular methods in the class of ROM that are used for reservoir modeling approximations are POD, TPWL, and DEIM. As discussed in Section 2.2, ROM methods project the exact model into a lower-dimensional subspace. The subspace basis in POD is achieved by accomplishing a singular value decomposition of a matrix containing the solution states obtained from previous runs [149]. POD has been implemented in different areas such as reservoir modeling [150], [151], finding the optimal control parameters in waterflooding [152], [153], and history matching [154]. Nevertheless, POD methods need to solve the full Jacobean of the matrix for projecting the non-linear terms in every iteration. Since the reservoir environment is highly non-linear, the speedup potential of POD to approximate the reservoir simulation is not significant. For instance, Cardoso et al. [151] achieved speedups of at most a factor of 10 for ROMs based on POD in reservoir simulation. To solve this drawback, retain the non-linear feature of parameters and further increase the speedup potential, a combination of the TPWL or DEIM method and POD has been the focus of attention in the literature. The combination of TPWL and POD was implanted in various cases such as waterflooding optimization [155], [156], history matching [157], [158], thermal recovery process [159], reservoir simulation [155], and compositional simulation [160]. In work carried out by Cardoso and Durlofsky [155], a POD in combination with TPWL could increase the speedup for the same reservoir discussed earlier from a factor of 10 to 450. Additionally, the application of DEIM and POD is applied in some studies to create proxies for reservoir simulation [161], [162], fluid flow in porous media [163], [164], and water flooding optimization [165]. Other methods to treat the non-linearity can be pointed out, such as Gauss-Newton with approximated tensors [166], truncated balanced realization [165], localized discrete empirical interpolation method [167], trajectory piecewise quadratic [168], and sparse proper orthogonal decomposition-Galerkin [169]. A comprehensive study of these methods in reservoir modeling can be found in work carried out by Suwartadi [170] and He [171].

2.4.3. Traditional Proxy Models (TPM)

In the literature, a wide variety of techniques can be considered as TPMs. This type of proxy can approximate different areas in the subsurface or surface environment such as production optimization [2], [172], uncertainty quantification [3], [173], history matching [4], [174], field development planning [175], risk analysis [176], [177], gas lift optimization [114], [178], gas storage management [179], screening purposes in fractured reservoirs [180], hydraulic fracturing [181], assessing the petrophysical and geomechanical properties of shale reservoirs [182], waterflooding optimization [183]–[186], well placement optimization [187]–[189], wellhead data interpretation [190], and well control optimization [191]. Additionally, TPMs have a wide range of applications in various EOR recovery techniques such as steam-assisted gravity drainage (SAGD) [192], CO_2-gas-assisted gravity drainage (GAGD) [193], water alternating gas (WAG) [194], [195], and chemical flooding [196].

2.4.4. Smart Proxy Models (SPM)

SPMs are implemented in various areas such as waterflood monitoring [10], [197], gas injection monitoring [11], and WAG monitoring [8] using the grid-based SPM, history matching [9], [12], and production optimization in a WAG process [8] using the well-based SPM. A brief summary of the PMs (including the TPMs and SPMs) used in the literature can be found in Table 2.

Table 2. Examples of PM applications in reservoir modeling.

Ref.	Subject	Sampling Technique	Underlying Model	Optimizer	Class
Kovscek and Wang [130]	Uncertainty quantification in a carbon dioxide storage case	–	Streamlines	–	MFM
Tanaka et al. [128]	Production optimization in waterflooding	–	Streamlines	GA	MFM
Wang and Kovscek [131]	History matching in a heterogeneous reservoir	–	Streamlines	–	MFM
Tang et al. [198]	Investigating the effects of the permeability heterogeneity and well completion in near-wellbore region	–	Streamlines	–	MFM
Kam et al. [133]	Three-phase history matching	–	Streamlines	GA	MFM
Taware et al. [135]	Well placement optimization in a mature carbonate field	–	Streamlines	–	MFM
Allam et al. [139]	History matching	–	Upscaling	–	MFM
Yang et al. [141]	Multiphase uncertainty quantification and history matching	–	Upscaling	–	MFM
Holanda et al. [146]	Reservoir characterization and history matching	–	CRM	–	MFM
Artun [147]	Characterizing interwell reservoir connectivity	–	CRM	–	MFM
Cardoso and Durlofsky [155]	Production optimization in waterflooding	–	TPWL/POD	Gradient-based	ROM
Xiao et al. [157]	History matching	Smolyak sparse [199]	TPWL/POD	Gradient-based	ROM
Rousset et al. [159]	Production prediction of SAGD operation	–	TPWL/POD	–	ROM
He and Durlofsky [160]	Compositional simulation of the reservoir	–	TPWL/POD	–	ROM
Gildin et al. [161]	Simulation of flow in heterogeneous porous media	–	DEIM/POD	–	ROM
Li et al. [163]	Compressible gas flow in porous media	–	DEIM/POD	–	ROM
Alghareeb and Williams [165]	Production optimization in waterflooding	–	DEIM/POD	–	ROM
Al-Mudhafar [2]	Production optimization in cyclic CO_2 flooding	–	PR, MARS, RF	–	TPM
Golzari et al. [172]	Production optimization in three different cases to increase recovery and net present value (NPV)	Adaptive LHS	ANN	GA	TPM
Amiri Kolajoobi et al. [173]	Uncertainty quantification and determination of cumulative oil production	LHS	ANN	–	TPM
Peng and Gupta [3]	Uncertainty quantification in a fluvial reservoir	Factorial	PR	–	TPM
Zubarev [4]	History matching and production optimization	LHS	PR, KG, ANN	GA	TPM
Guo et al. [174]	History matching in a channelized reservoir	Random Selection	SVR	Distributed Gauss-Newton [200]	TPM
Avansi [175]	Field development planning	BBD	PR	–	TPM
Ligero et al. [176]	Risk Assessment in economic and technical parameters on an offshore field	Factorial	PR	–	TPM

Risso et al. [177]	Assessment of risk curves for uncertainties in the reservoir	BBD, CCD	PR	–	TPM
Ghassemzadeh and Charkhi [178]	Gas lift optimization to maximize recovery and NPV	–	ANN	GA	TPM
Ebrahimi and E. Khamehchi [114]	Gas lift optimization in NGL process	LHS	SVR	PSO, GA	TPM
Zangl et al. [179]	Gas storage management and optimization for pressure	Factorial	ANN	GA	TPM
Artun et al. [180]	Screening and optimization of cyclic pressure pulsing in naturally fractured reservoirs	–	ANN	GA	TPM
Gu et al. [183]	Waterflooding optimization in terms of watercut	–	XGboost	DE	TPM
Chen et al. [184]	Waterflooding optimization in terms of recovery and NPV	LHS	KG	DE	TPM
Ogbeiwi et al. [185]	Optimization of water injection rate and oil production rate in waterflooding	BBD	PR	GA	TPM
Bruyelle and Guérillot [186]	Waterflooding optimization in terms of well parameters	BBD	ANN	Covariance Matrix Adaptation Evolution Strategy [201]	TPM
Bruyelle and Guérillot [187]	Well placement Optimization to maximize recovery and NPV	BBD	ANN	Covariance Matrix Adaptation Evolution Strategy	TPM
Hassani et al. [188]	Optimization the horizontal well placement	Optimal, LHS	PR, RBF	GA	TPM
Nwachukwu et al. [189]	Injector well placement optimization to maximize recovery and NPV	Random Selection	XGboost	–	TPM
Aydin et al. [190]	Monitoring of a geothermal reservoir temperature and pressure from wellhead data	–	ANN	–	TPM
Wang et al. [191]	Well control optimization to maximize recovery and NPV	LHS	SVR	Non-dominated sorting GA-II [202]	TPM
Simonov et al. [203]	Production optimization in a miscible flooding case	LHS	RF	MC	TPM
Redouane et al. [204]	Well placement optimization to maximize recovery	LHS, Sobol, Halton	ANFIS [205]	GA	TPM
Fedutenko et al. [192]	Production prediction of SAGD operation	LHS	PR, KG, RBF	–	TPM
Al-Mudhafar and Rao [193]	Recovery evaluation in CO_2-GAGD operation	LHS	PR, MARS, GBM	–	TPM
Jaber et al. [194]	Recovery evaluation in miscible CO_2-WAG flooding	BBD	PR	–	TPM
Agada et al. [195]	Recovery and net gas utilization factor optimization of a CO_2-WAG operation in a fractured reservoir	BBD	PCE	GA	TPM
Elsheikh et al. [206]	Watercut determination in waterflooding cases	Nested Sampling, MCMC	PCE	–	TPM
Yu et al. [207]	History matching and production forecasting	Hammersley [208]	GP	–	TPM

Kalla and White [209]	Optimization of a gas well with water conning	OAS	PR	–	TPM
Ibiam et al. [196]	Sensitivity analysis and polymer flooding optimization	LHS	PR	PSO	TPM
Kim and Durlofsky[210]	History matching and well-by-well oil and water flow rate prediction in waterflooding	Random selection	RNN	PSO	TPM
Kim and Durlofsky [211]	Predict NPV with time-varying BHP	Uniform distribution	RNN	PSO	TPM
Kim et al. [212]	Multi-well placement optimization	Uniform distribution	CNN	PSO	TPM
Haghshenas et al. [10]	Evaluating the effect of injection rates on oil saturation using the grid-based SPM	LHS	ANN	–	SPM
Alenezi and Mohaghegh [197]	Evaluating the effect of injection rates on oil saturation and pressure using the grid-based SPM	Random Selection	ANN	–	SPM
Amini and Mohaghegh [11]	Gas injection monitoring in porous media using the grid-based SPM	–	ANN	Gradient descent	SPM
Gholami et al. [8]	WAG monitoring and production optimization using the grid-based and well-based SPMs	LHS	ANN	–	SPM
He et al. [9]	History matching using well-based SPM	LHS	ANN	DE	SPM
Shahkarami et al. [12]	History matching using well-based SPM	LHS	ANN	–	SPM
Ng et al. [13]	Production optimization in a fractured reservoir	–	ANN	PSO	SPM

ANFIS: adaptive neuro fuzzy inference system; **ANN:** artificial neural networks; **BBD:** Box-Behnken design; **CCD:** central composite design; **CNN:** convolutional neural networks; **CRM:** capacitance-resistance modeling; **DE:** differential evolution; **DEIM:** discrete empirical interpolation method; **GA:** genetic algorithm; **GBM:** gradient boosting machine; **GP:** genetic programming; **KG:** kriging; **LHS:** Latin hypercube sampling; **MARS:** multivariate adaptive regression splines; **MC:** Monte Carlo; **MCMC:** Markov chain Monte Carlo; **MFM:** multi-fidelity model; **NPV:** net present value; **OAS:** orthogonal array sampling; **PCE:** polynomial chaos expansion; **POD:** proper orthogonal decompositions; **PR:** polynomial regression; **PSO:** particle swarm optimization; **RBF:** radial basis functions; **RF:** random forest; **RNN:** recurrent neural networks; **ROM:** reduced-order model; **SPM:** smart proxy model; **SVR:** support vector regression; **TPM:** traditional smart model; **TPWL:** trajectory-piecewise linear; **XGBoost:** extreme gradient boosting.

2.5. Conclusions

The most significant advantage of constructing a proxy model is the reduction in computational load and the time required for tasks such as uncertainty quantification, history matching, or production forecasting and optimization. According to the literature, different classes of proxy models exist, and there is no agreement on the proxy model categorization. Existing categories do not provide a comprehensive overview of all proxy model types with their applications in the oil and gas industry. Furthermore, a guideline to discuss the required steps to construct proxy models is needed.

In this review, different classes of proxy models are discussed, and a new classification based on the development strategy is proposed. The proxy models in this classification fall into four groups: multi-fidelity, reduced-order, traditional proxy, and smart proxy models. The methodology for developing the multi-fidelity models is based on simplifying physics, and reduced-order models are based on the projection into a lower-dimensional. The procedure to develop traditional and smart proxy models is mostly similar, with some additional steps required for smart proxy models. Smart proxy models implement the feature engineering technique, which can help the model to find new hidden patterns within the parameters. As a result, smart proxy models generate more accurate results compared to traditional proxy models. Different steps for proxy modeling construction are comprehensively discussed in this review. For the first step, the objective of constructing a proxy model should be defined. Based on the objective, the related parameters are chosen, and sampling is performed. The sampling can be either stationary or sequential. Then, a new model is constructed between the considered inputs and outputs. This underlying model may be trained based on statistics, machine learning algorithms, simplifying physics, or dimensional reduction. For optimization purposes, this work describes some of the popular stochastic optimizers as a tool to couple with the proxy models. Finally, the application of various proxy models in oil and gas and reservoir modeling for each category is presented in this paper.

This review paper provides a comprehensive guideline to develop proxy models. This guideline provides a better, structured, and more efficient approach to help model, optimize, and forecast more complex problems in future studies. Additionally, this paper provides the reader with a better understanding of the different proxy model categories, and it provides various applications for the proxy models in the oil and gas industry.

Chapter 3. Screening of Waterflooding Using Smart Proxy Model Coupled with Deep Convolutional Neural Network

Preface

A version of this chapter has been published in the Geoenergy Science and Engineering Journal. I am the primary author, along with the co-author Dr. Lesley A. James. The detail of the author's contributions is as follow:

Peyman Bahrami: Conceptualization, Methodology, Writing–original draft, and Writing–Review & Editing, Visualization

Dr. Lesley A. James: Conceptualization, Methodology, Writing–Review & Editing, and Supervision

Reference: P. Bahrami and L. A. James, "Screening of waterflooding using smart proxy model coupled with deep convolutional neural network," *Geoenergy Science and Engineering*, vol. 221, p. 111300, Feb. 2023, doi: 10.1016/j.petrol.2022.111300.

Abstract

The objective of this work is to identify any efficiency and accuracy improvements in smart proxy modeling (SPM). SPM is a novel methodology which include additional steps in the construction process compared to traditional proxy models (TPM). We discuss the advantages of SPM compared to TPM where SPM implements feature engineering techniques which involves generating new static and dynamic parameters. The new extracted parameters help the model to capture hidden patterns within the parameters, which eventually increase the accuracy of SPMs compared to TPMs.

Based on our literature review, we target our investigation into techniques to improve efficiency and accuracy by focusing on sampling (sequential sampling), feature ranking, and underlying model construction (CNN instead of ANN). In existing SPM literature, only one technique is used during each construction step where there are opportunities to explore novel construction steps to improve overall SPM accuracy and efficiency. Sequential sampling helps to construct the SPM with the lowest number of high-fidelity model execution and it avoids resampling, thereby saving time and making the SPM workflow more efficient. The average feature ranking technique described in this work provides a more confident prioritization of input parameters which eventually helps the overall efficiency in the feature selection step. CNN model as the underlying model provides higher accuracy than implemented ANN models in literature. The SPM with ANN underlying model provides an accuracy of 89-92% compared to the 99% and 94% of the CNN technique for the pressure and oil saturation predictions, respectively.

In this paper, we construct a grid-based SPM of a Norwegian offshore field undergoing waterflooding. The designed parameters for this case are the individual liquid rate of the producers, and the outputs are individual grid's oil saturation and pressure. It is shown that the final results for screening purposes generated by SPM can confidently be used to mimic the behavior of the numerical models. Also, the results of feature ranking illustrate that some of the extracted data used in the SPM construction steps influence the model's outputs, confirming SPM capability.

Keywords: Smart proxy model; Convolutional neural network; Average feature ranking; Sequential sampling; Waterflooding screening; Machine learning

3.1. Introduction

Waterflooding is the most widely and economically applied secondary recovery method. There are two main reasons to perform waterflooding: (1) to keep the pressure above the saturation pressure and prevent gas production, and (2) to sweep the oil toward the producers and have a higher ultimate oil recovery [213]. It is always a good practice to screen the waterflooding process using the numerical models to fulfill these goals. A good screening plan, or observation of the changes in grids' oil saturation and pressure, significantly increases the required knowledge for any field development decisions. Hence, to have a proper screening for the reservoir, a large number of numerical model runs need to be performed. However, this task is challenging in terms of computational time and manpower.

Using a proxy model (PM) is an alternative way to speed-up getting the desired results from the numerical model. PM is a substitute or an approximation of a numerical model, mathematical model, or a simulator which generates the results in a reasonable accuracy much quicker. PMs can be categorized based on different aspects such as time dependency, approximation strategy, or objective, but one of the main categorizations is the one done by Eldred and Dunlavy [15]. They grouped PMs into data-fit, multi-fidelity, and reduced-order types. Data-fit models implement a regression on the data obtained from the original high-fidelity model, and they usually use a machine learning method as the underlying model. Polynomial regression, artificial neural networks, and kriging are a few examples that fall into this category. In multi-fidelity models, a high fidelity model is simplified into an easier model through simplifying assumptions [32] or coarser discretization [31]. Streamline modeling, upscaling, and capacitance-resistance modeling (CRM) are the most popular techniques of multi-fidelity models in reservoir modeling. Reduced-order models (ROM) project the high-fidelity model into a lower-dimensional subspace with equivalent characteristics by neglecting the irrelevant parameters. Some popular techniques in this class are proper orthogonal decompositions (POD) [34], trajectory-piecewise linear (TPWL) [35], and discrete empirical interpolation method (DEIM) [36].

In the literature, each of these classes was previously used in various areas of reservoir engineering. Data-fit models were applied on production optimization [2], [172], [214], uncertainty quantification [3], [173], history matching [4], [174], [207], field development planning [175], risk analysis [176], [177], gas lift optimization [114], [178], gas well optimization [209], gas storage management [179], screening purposes in fractured reservoirs [180], waterflooding optimization [183]–[186], well placement optimization [187]–[189], wellhead data interpretation [190], and well control optimization [191]. Also, they had a wide application in various enhanced oil recovery (EOR) techniques such as steam-assisted gravity drainage [192], CO_2-gas assisted gravity drainage [193], water alternating gas (WAG) [194], [195], and chemical flooding [196]. Multi-fidelity models were applied in production optimization [128]–[130], history matching [131]–[134], and well placement optimization [135] through streamline modeling, and upscaling is used in reservoir modeling objectives [139]–[142]. CRM as another class of multi-fidelity model is implemented in history matching and production forecasting [146], [147]. In the ROM category, POD was implemented in different areas such as reservoir modeling [150], [151], control parameters optimization in water flooding [152], [153], and history matching [154]. The use of TPWL was tested in various cases such as waterflooding optimization [155], [156], history matching [157], [158], thermal recovery process [159], reservoir simulation [155], and compositional simulation [160], and DEIM is applied in reservoir simulation [161], [162], fluid flow in porous media [163], [164], and water flooding optimization [165].

Another classification method for the PMs is the one done by Mohaghegh [7]. He divided PMs into traditional proxy models (TPM) and smart proxy models (SPM). TPM involves using mathematically simplified and statistically driven models such as ROMs, multi-fidelity, and data-driven models. PMs in this work are classified into four classes of multi-fidelity models, ROMs, TPMs, and SPMs based on their different development strategies [215]. TPM in this work refers to the PMs that use a machine learning method as the underlying model (is the same class as data-fit PMs). These models have shortcomings in accuracy when they reproduce the results obtained from numerical models, especially in complex cases. SPMs have additional steps in the construction process compared to TPMs, and it is needed to form a new dataset with new parameters to uncover the hidden patterns within the problem for the underlying model. SPMs are grouped as grid-based or well-based. A grid-based SPM reproduces the results at the grid-level, and it is suitable for problems where it is desired to monitor an output parameter in grid blocks. The grid-based SPM has been implemented for waterflood monitoring [10], [197] and gas injection monitoring [11], WAG monitoring [8]. A well-based SPM generates the results at the well-level, and it applies to cases when having an estimate of the production is essential such as history matching [9], [12] and production optimization [8]. The existing literature regarding the SPM used the same technique in different steps of SPM construction [8]–[12], [197]. For example, only one type of sampling technique (stationary Latin hypercube sampling), one feature ranking technique (ANN ranking), and one underlying model (ANN) are implemented in the models.

Our objective is to identify efficiency and accuracy improvements in SPM modeling. Specifically, we investigate the use of sequential sampling to avoid the need to repeat the whole procedure using resampling where we can possibly augment more sample points to the existing design to save the time of resampling, generation of data sets, and the rest of the SPM workflow. While existing literature does not specifically say how they selected and ranked input parameters in the SPM, average feature ranking technique is introduced in this work. This technique helps to have a more confident prioritization of input parameters to do the feature selection which eventually helps the overall efficiency. We also investigate the use of CNN instead of ANN to predict oil saturation and pressure profiles in grid-based modeling. In this paper, a grid-based SPM for the Volve reservoir model is constructed, and it is used to screen the oil saturation and pressure of grids during a waterflooding process. Section 3.2.3 discusses the sampling step in more detail, Section 3.2.4 introduces the average feature ranking that is used in this paper, Section 3.2.5 describes the CNN model in detail, and Section 3.3 introduces the Volve case study used in this paper.

3.2. Methodology

This work tries to show the differences between a TPM and an SPM in the construction process. A TPM is usually constructed by the following steps: (1) define the objective, input, and output parameters, (2) perform the sensitivity analysis (if needed), (3) sampling, (4) run the high-fidelity model to extract only the output parameter, (5) train and validate the underlying model, and (6) employ the TPM to obtain the results. These steps are discussed in Sections 3.2.1 to 3.2.5 in more detail.

The SPM has additional development steps compared to the TPM. Figure 4 presents the steps required to construct a TPM and an SPM. After sampling and running the simulator (or high-fidelity model) at the sample points, a combination of static and dynamic parameters needs to be extracted from the simulator. Then, these parameters are used to form a new dataset. Finally, an underlying model trains and validates between the inputs and outputs to complete the SPM construction. The constructed SPM can then predict the output parameter in a much shorter time for the initially selected inputs and is applicable to fulfill the objective of the problem. In general, if the aim is to run the simulator a few times, constructing an SPM is not recommended. However, if we desire to run the simulator many times, the SPM construction can be quite helpful.

This paper presents the construction of a grid-based SPM. Rows in the new dataset correspond to individual grids of the numerical reservoir model, and the columns are the static and dynamic parameters extracted at different timesteps. The parameters forming the columns could be a wide variety of parameters such as the grids' location, geometry, properties, or distances to the objects such as wells, boundaries, etc. Extracting the parameters for the individual grids sometimes forms a massive dataset. For example, in work done by Gholami et al. [8], the grid-based dataset initially consisted of 396,000 rows and over 1000 parameters.

Constructing a TPM on such a big dataset can be challenging and time-consuming. On the other hand, one of the main advantages of using an SPM is that it only takes a few seconds to approximate the full reservoir. If the time required to construct an SPM is significant, the applicability of the SPM diminishes, and it is known as the curse of dimensionality [14], [61]. So, it is crucial to do some preprocessing to reduce the number of rows (observations) and columns (parameters) of this dataset. Also, performing the sampling in an effective procedure is needed to save time on the construction of the SPM, which can be fulfilled through sequential sampling used in this research.

3.2.1. Defining Objective, Inputs, Outputs and Ranges

To construct a PM, one must first determine the objective and the dependent inputs and outputs. Choosing the proper inputs and outputs may be the most important step in PM construction. The objective of a reservoir engineering problem could be production optimization, history matching, uncertainty analysis, and many others. In each of the objectives, some input and output parameters are effective. For example, suppose the objective is the production optimization in a WAG injection case. In that case, the inputs can be water or gas injection rates, water viscosity, cycle length, well production rates, etc., and the output can be the total oil production rate, watercut, or gas-oil ratio. In this study, the objective is to screen the oil saturation and pressure during a waterflooding case. The effective parameters, in this case, are the individual liquid production rate of wells.

3.2.2. Sensitivity Analysis

A reservoir environment is considered a non-linear and complex problem in which many parameters play a role. Dealing with many parameters in such an environment and running the simulator based on all the parameters sometimes is very costly and time-consuming. In such a case, sensitivity analysis can find the non-influential inputs and reduce their number. So, sensitivity analysis is essential in constructing the PM in cases with a high number of inputs and is performed before the sampling step [46], [47]. There are only five inputs under investigation in this paper, and sensitivity analysis is not performed.

3.2.3. Sampling

Sampling is defined as the procedure to generate datapoints in the domain range of the input parameters. Usually, a higher number of sample points over the search space results in a more accurate PM (or SPM). However, as the number of sample points increases, the PM's construction time increases, and picking a small number of sample points is required to avoid falling into the curse of dimensionality. Hence, the main objective of sampling is to get maximum information by generating the minimum possible sample points. Therefore, choosing a proper sampling strategy is a crucial construction step in the construction of a PM.

Sampling techniques are divided into stationary and sequential methods. Stationary sampling techniques seek to evenly distribute the sample points over the domain of the inputs, and they use a fixed pattern in all the problems under study [63]. Factorial designs (full and factorial designs), optimal designs, Latin hypercube sampling (LHS), orthogonal array sampling, and random sampling such as Monte Carlo are examples of this technique.

Stationary sampling is easy to employ; however, if the PM was not approved in any stages during construction (mainly in the validation stage) the whole procedure for SPM construction from sampling stage needs to be repeated. This shortcoming can cause an increase in the computation cost associated with constructing a new PM with a higher number of sample points. This problem is even more significant in non-linear and complex problems such as the reservoir environment involving many parameters. Constructing an SPM also suffers the disadvantage of the stationary sampling techniques. The number of sample points (accordingly, the required runs) depends on different geological properties and operational constraints, and there is no rule of thumb to select the suitable number of sample points. For example, in the research done by Gholami et al. [8], the number of sample points were 13 for the four designing parameters in a WAG process, or in work done by He et al. [9], only three sample points were picked for the two design parameters to construct an SPM for a history matching problem. Hence, using an alternative sampling technique, such as sequential sampling, could be beneficial.

In sequential sampling, the design starts with initial sample points, and if the constructed PM does not achieve a good accuracy, new sample points are sequentially added to the existing ones. This process stops whenever the model reaches a preferred performance and accuracy [72].

Hence, PM construction time is significantly decreased using sequential sampling techniques, and it avoids repeating the sampling and running the simulator [73]. Workflows for stationary and sequential sampling techniques are presented in Figure 5.

In this paper, sequential LHS sampling technique is used, which is a novel procedure in the SPM's construction. There are various types of algorithms to generate the initial sample points and the augmented points.

3.2.3.1. Random LHS

LHS is a stratified sampling technique that divides each parameter into N bins or equal intervals. Then, it distributes the sample points in each bin of the parameters. In LHS, it is possible to choose any number of sample points, and splitting the parameters domain happens based on that number. So, there is a control over the number of sample points, and it is a big advantage in constructing the PMs where we are looking to limit the number of simulator runs [69]. Not all Latin hypercube designs uniformly distribute the sample points in the domain, and it is necessary to optimize the space-filling procedure. For example, Figure 6 shows an unfavorable LHS design in which major parts of the space are not filled with sample points. Viana et al. reviewed some of the techniques to optimize the LHS to better fill up the domain under study [70].

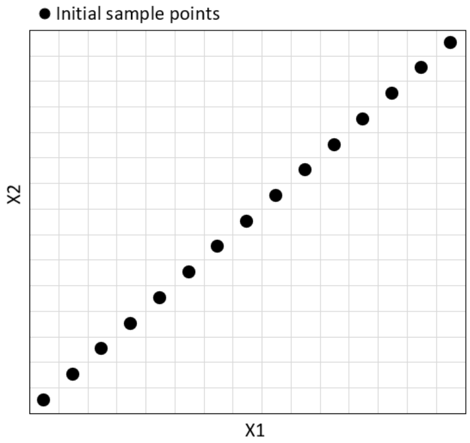


Figure 6. An example of an unfavorable LHS design

In random LHS, the sample points are randomly placed in the intervals so that they only repeat in the intervals of each parameter once. Figure 7a shows an example of the random LHS design for two parameters with 15 sample points. As can be seen, the sample points are not uniformly distributed in the parameters' domain, and there are big spaces that are not covered by sample points. Simple augmentation or optimized augmentation can be implemented if more sample points are required to be added to the initial design. In simple augmentation, it is not guaranteed that the new points keep the distance from the existing points; however, the optimized augmentation tries to keep the distance from the existing points. In Figures 7b and 7c, five more points are added to the initial design using the simple and optimized augmentation techniques.

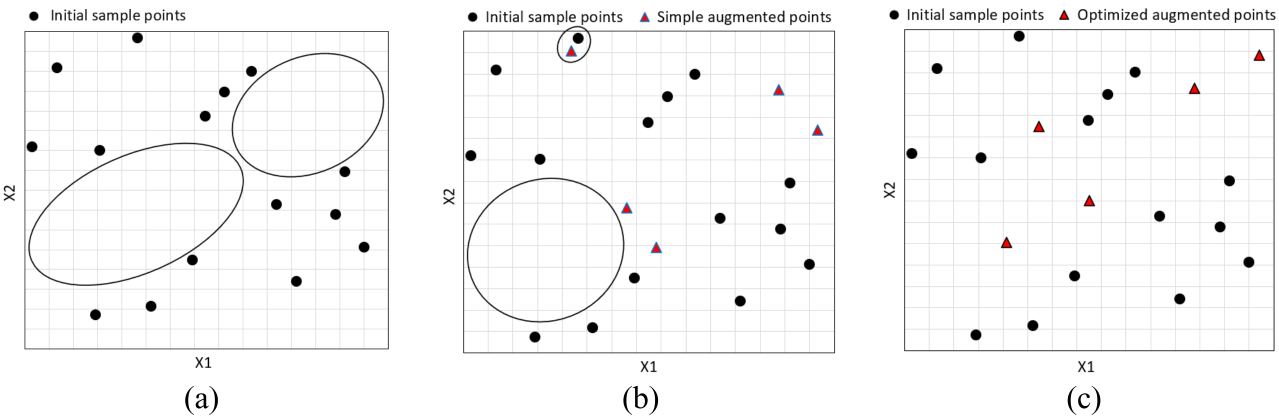
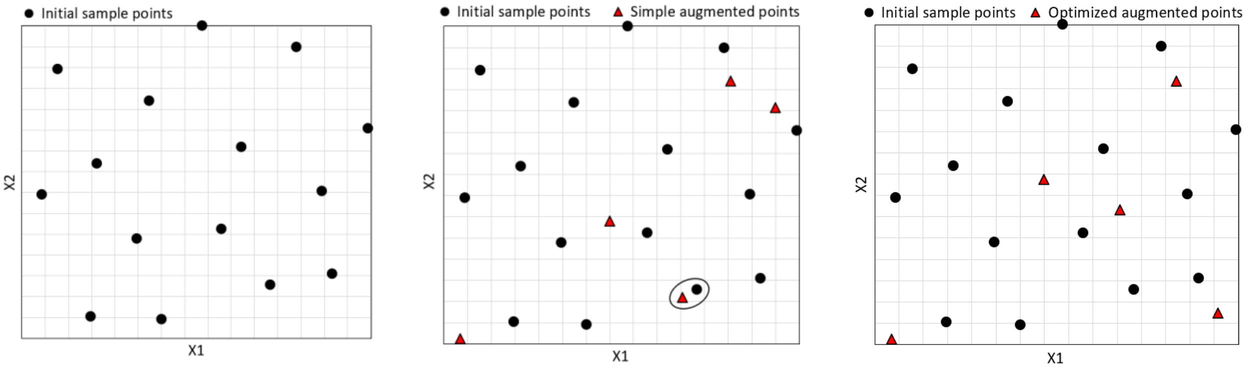


Figure 7. Random LHS design: (a) initial design, (b) simple augmented, and (c) optimized augmented points

3.2.3.2. Optimal LHS

This sampling technique seeks to create an LHS design that uniformly distributes the initial sample points in the parameters' domain. The algorithm for the optimal design is based on the columnwise pairwise with respect to the S-optimality criterion. S-optimality distributes the sample points by maximizing the harmonic mean distance from other sample points [216]. Figure 8 shows the initial design and the augmented points in an optimal LHS.

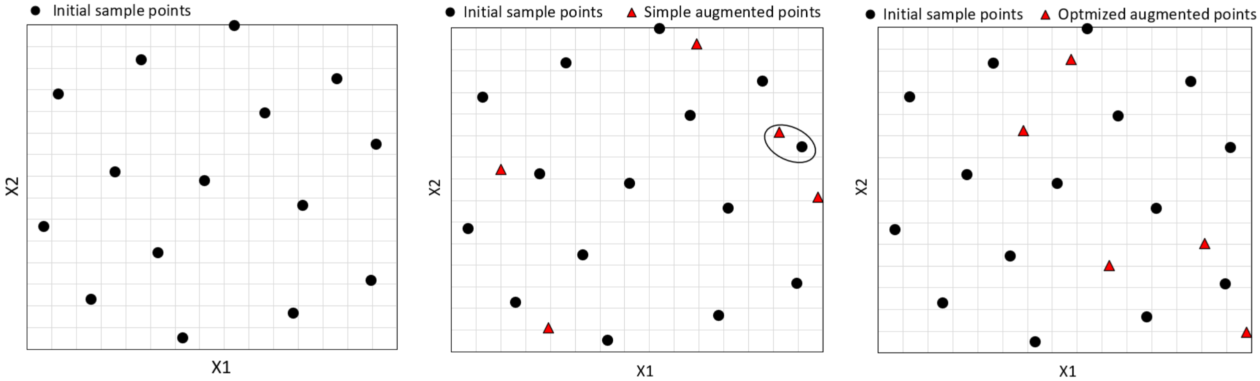


(a) (b) (c)

Figure 8. Optimal LHS design: (a) initial design, (b) simple augmented, and (c) optimized augmented points

3.2.3.3. LHS with a Genetic Algorithm

This sampling technique uses genetic algorithm (GA) to better distribute sample points in an S-optimality criterion. An example of the sample points and the augmented points is demonstrated in Figure 9.



(a) (b) (c)

Figure 9. LHS with GA: (a) initial design, (b) simple augmented, and (c) optimized augmented points

In this work, a genetic algorithm coupled with LHS is used to create the initial sample points, and an optimized algorithm is used to augment more sample points.

3.2.4. Dataset Generation, and Feature Selection

After setting the sample points, it is necessary to run the high-fidelity model (here it is the simulator) at the sample points. Then various static and dynamic parameters are extracted from the simulator runs and forms a new dataset. In this dataset, the number of columns equals the number of parameters, and the number of rows is the multiplication of grid numbers and the run numbers

To justify that extraction of more data and forming a new dataset is important in SPM construction, the influence of the extracted parameters on the output is tested using the feature ranking technique. Feature ranking is the process of finding the importance of input parameters on the output parameter in the dataset. There are different feature ranking algorithms such as linear machine learning, decision tree, and permutation techniques [217]. In linear machine learning techniques, a regression model fits a dataset, and a set of coefficients for parameters is obtained. Hence, the model's prediction is a weighted sum of the input parameters, and the coefficients can be used further for parameter importance scoring. Linear regression, ridge regression, logistic regression, or elastic net are examples of this category. Decision tree techniques try to do the feature ranking by reducing the criterion to choose the split points in a tree or an ensemble of trees. Random Forest (RF) or gradient boosting algorithms fall in this class of feature ranking methods. In permutation techniques, the algorithm takes a parameter at a time and tests the model's performance by applying random noise to the parameter. So, the algorithm can determine how much the model depends on the individual parameters.

The importance of parameters can be used to reduce the dimensions of the dataset, which is called feature selection. Feature selection speeds up data collection, decreases training time, and lowers the overfitting chance. There are three classes of feature selection; filter, wrapper, and hybrid. The filter class employs a statistical factor which scores the relationship of the individual input parameters to the output parameter. Then, the input parameters are ranked, and the method selects the best subset of input parameters. Different algorithms can be used in the wrapper class that iteratively search different input parameters' combinations to find the optimal subset. The wrapper methods have the advantage of selecting the optimal set from all the available input parameters [218]. Although wrapper methods are accurate, they are computationally more expensive compared to filter methods, and they might overfit. As a result, for a dataset with a large number of input parameters, using the standalone wrapper methods would not be recommended. Instead, hybrid methods, which are a combination of filter and wrapper methods, are more applicable. Most recent hybrid methods initially reduce the number of input parameters using one of the filter methods, then they employ an expensive wrapper method [219].

Different results of variable importance and ranking would be achieved depending on the choice of predictive models for the feature ranking approach [220]. Hence, it is always a challenging decision to choose a proper model in feature ranking to be further used for feature selection. In this work, *fscaret* R package [221] is used to rank and reduce the input vector. *fscaret* is a filter method that generates a data frame of variable importance, errors of the input parameters, and the gradient of parameter importance by combining different algorithms. In this work, a combination of 23 different models is compared in *fscaret* and averaged to provide a more stable variable ranking. Then, we used the gradient of importance to select different subsets of input parameters. The gradient of importance was calculated using Equation 11.

$$gradient\ of\ importance = \frac{absolute\ difference\ of\ current\ and\ previous\ importance}{previous\ importance} \times 100 \qquad (11)$$

The list of applied models for feature ranking and feature selection is presented in Table 3.

Table 3. Different predictive models used for feature ranking

1	random forest (RF)	13	polymerase chain reaction (PCR)
2	penalized regression	14	partial least squares (PLS)
3	binary search tree (BSTREE)	15	fits elastic net regression (ENET)
4	neural network	16	cubist
5	Bayesian generalized linear model (GLM)	17	multilayer perceptron with weight decay
6	gradient boosting machine (GBM)	18	multilayer perceptron
7	weighted K-nearest neighbors (KKNN)	19	ridge regression
8	K-nearest neighbors (KNN)	20	relaxed lasso
9	least angle regression (LARS)	21	projection pursuit regression (PPR)
10	feed-forward neural networks and multinomial log-linear	22	partial least squares regression for generalized linear models (PLSRGLM)
11	Lasso regression	23	recursive partitioning and regression trees (RPART)
12	neural networks with a principal component step (PCANNET)		

3.2.5. Training and Validating the Underlying Model

The next step after forming the grid-based dataset includes training a machine learning model on the data. In this research, a one-dimensional convolutional neural network (1D-CNN) model was constructed. Using the 1D-CNN model is a novel way to build the underlying model of the SPM not previously used in the literature.

CNN is a type of deep neural network primarily designed for the analysis of 2D image data. However, in recent years, its usage has been extended to 1D (such as time series and natural language processing) and 3D data (such as video) [222]. CNN is a generalized technique that works well in big datasets with many parameters. In common artificial neural networks, if the dataset contains many parameters, the training time would become significant if the number of hidden layers increases [223] whereas CNN trains more quickly as it uses the filters to reduce the size of dataset. CNN structure generally involves using three main layers of convolutional, pooling, and fully connected layers. In the convolution layer, multiple filters are applied to the inputs. The filter is a sliding window that can move in different directions, and it extracts parameters from the dataset. The filter applies to a part of the dataset (or image or videos) by taking a dot product between the filter and that part of the dataset. The CNN is referred to as 3D, 2D, and 1D if the filter moves in three, two, and one direction(s), respectively (Figure 10).

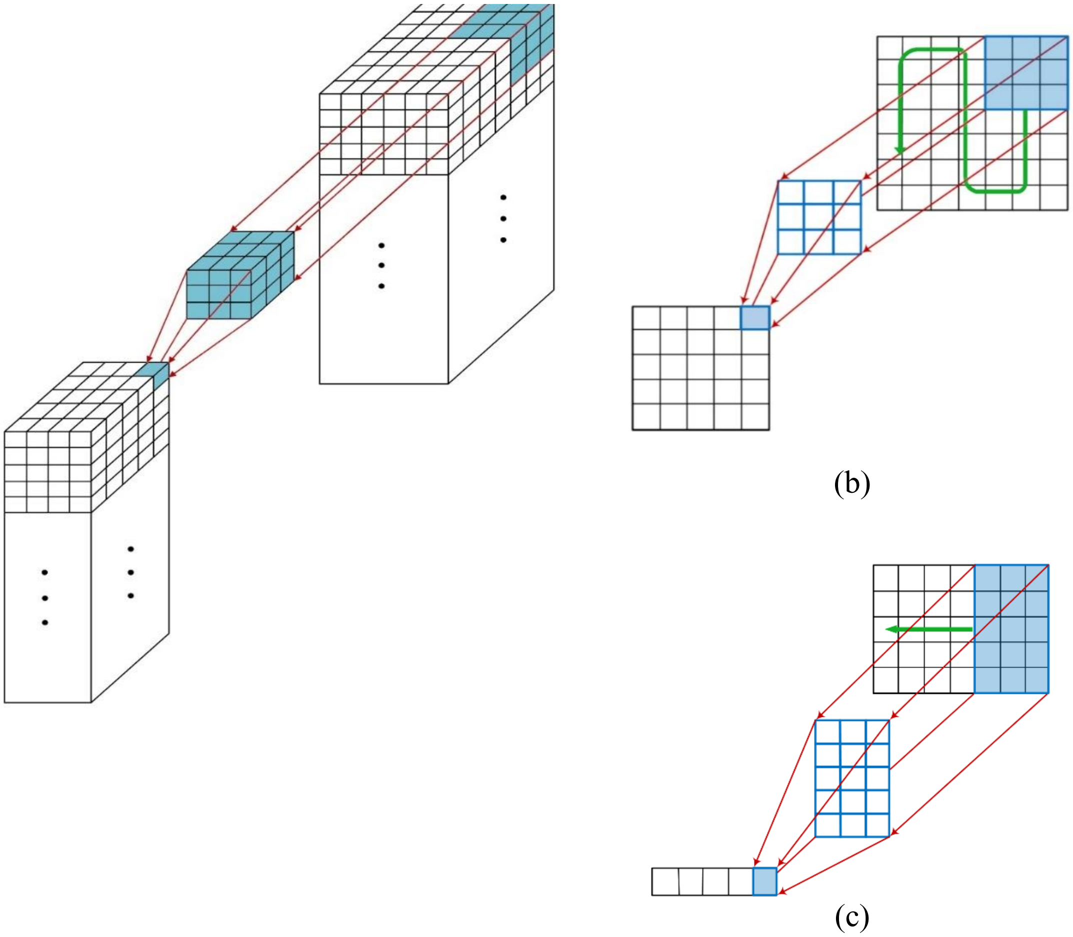


Figure 10. Illustration of movement in: (a) 3D-filter, (b) 2D-filter, and (c) 1D-filter

The pooling layer reduces the dimensionality and selects the most significant parameters through subsampling. Max-pooling, min-pooling, and average-pooling are the most common pooling operations in this layer. When different filters are applied to the input data, the result is flattened and goes through the fully connected layer (dense layer) or a multi-layer perceptron network (MLP). This fully connected layer consists of neurons, weights, and biases in which the neurons in one layer are connected to the neurons in the next layer and finally to the output layer. In the model used in this research for the grid-based model, the convolution layer consists of three filter sublayers; the fully connected layer includes two hidden sublayers and no pooling layer. A schematic of a 1D-CNN with one filter layer and one dense layer is shown in Figure 11.

There are some important parameters associated with each CNN model that need to be designed, such as filter size, kernel size, neuron size, and the number of sublayers in the convolutional layer or fully connected layer. Filter size refers to the number of filters in each filter sublayer, and kernel size is the width of the sliding filter window. These parameters need to be designed to train a more accurate model. The design procedure started by choosing the number of sublayers in the convolutional and fully connected layers. Various sublayers were tested and eventually, three filter sublayers and two hidden layers were selected as the base structure of the 1D-CNN model. Then, a design of the experiment was performed on different adjusting parameters based on the D-optimal method.

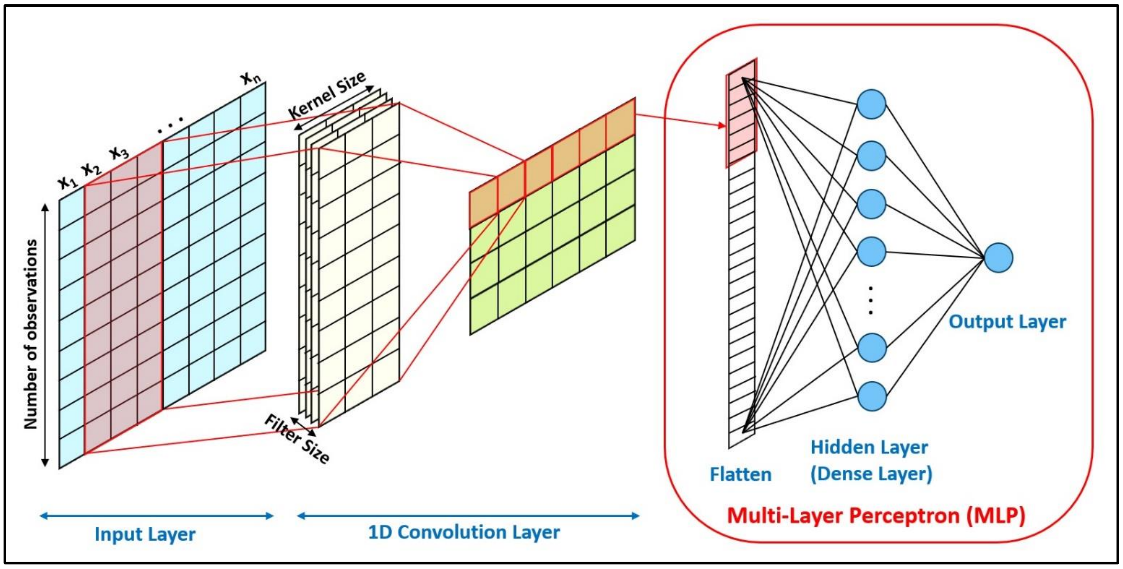


Figure 11. 1D-CNN schematic with one filter layer and one dense layer

After training the underlying model, the goodness of fit was evaluated using Root Mean Squared Error (RMSE) and the coefficient of determination (R^2). The model that has a lower RMSE and higher R^2 can be considered a reliable model. RMSE and R^2 are calculated based on Equations (12) and (13).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(y_{prd,i} - y_{act,i}\right)^2}{n}} \tag{12}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_{act,i} - y_{prd,i}\right)^2}{\sum_{i=1}^{n}\left(y_{act,i} - y_m\right)^2} \tag{13}$$

where y_{act} is the actual value, y_{prd} is the predicted value, i is the data record number, y_m is the average of the experimental value, and n is the total number of records. Figure 12 demonstrates the SPM construction workflow used in this work.

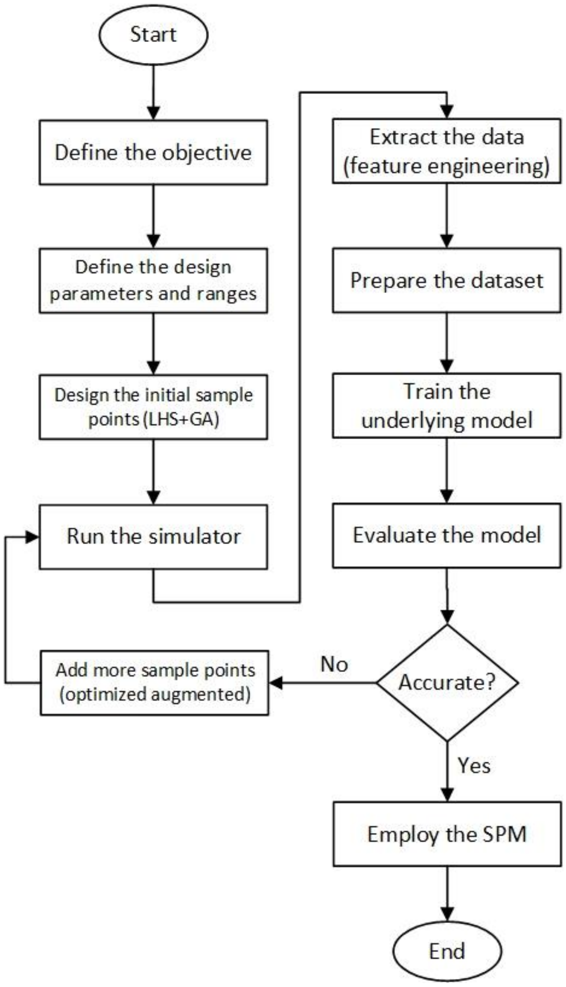

Start
Define the objective
Define the design parameters and ranges
Design the initial sample points (LHS+GA)
Run the simulator
Extract the data (feature engineering)
Prepare the dataset
Train the underlying model
Evaluate the model
Accurate?
No
Add more sample points (optimized augmented)
Yes
Employ the SPM
End

Figure 12. Workflow for the presented SPM methodology

3.3. Case Study

In this work, an SPM for the Volve reservoir simulation model is constructed. The Volve field was discovered in 1993, and it is an offshore oil field located 200 km west of Stavanger and 5 km north of the Sleipner Øst field. The Volve field is a 2×3 km, fault-delimited structure with a total proven 27.5 million Sm^3 of oil that is expected to produce 11.4 million Sm^3 oil and 1.5 billion Sm^3 rich gas. The field consists of a sandstone reservoir with 93% net-to-gross ratio, porosity of 21%, well test permeability of 1 Darcy, and average water saturation of 20% in the oil zone. The developed reservoir simulation model for this field uses a reservoir pressure and temperature of 340 bar and 110 °C at 3060 m, respectively. The gas-oil ratio varies between 111 and 157 Sm^3/Sm^3, and the formation volume factor varies between 1.33 and 1.45 m^3/Sm^3. Based on the capacities of the process plant on the jack-up platform, the field has limitations of 16,000 Sm^3/d for injected water, 9000 Sm^3/d for oil production rate, and 10,400 Sm^3/d for water production rate. The field started to produce oil in February 2008, and it was under waterflooding recovery until it was shut in September 2016. Volve produced a total of 9.98 million Sm^3 oil. Figure 13 shows the location of Volve field, and Figure 14 illustrates the reservoir simulation model. The reservoir model in this study has three injectors and five producers. The location and trajectory of the wells are presented in Figure 15.

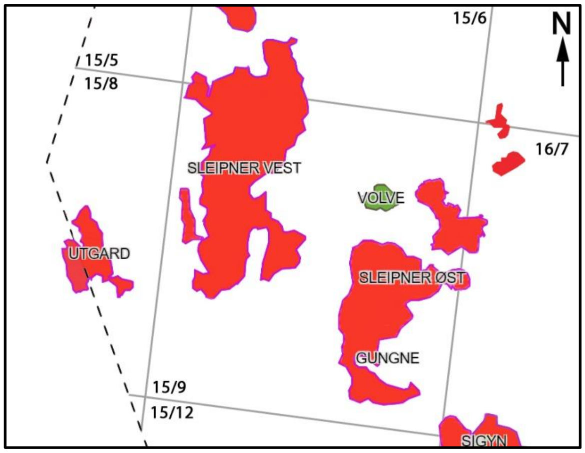


Figure 13. Volve field location (modified from https://www.norskpetroleum.no/en/facts/field/volve)

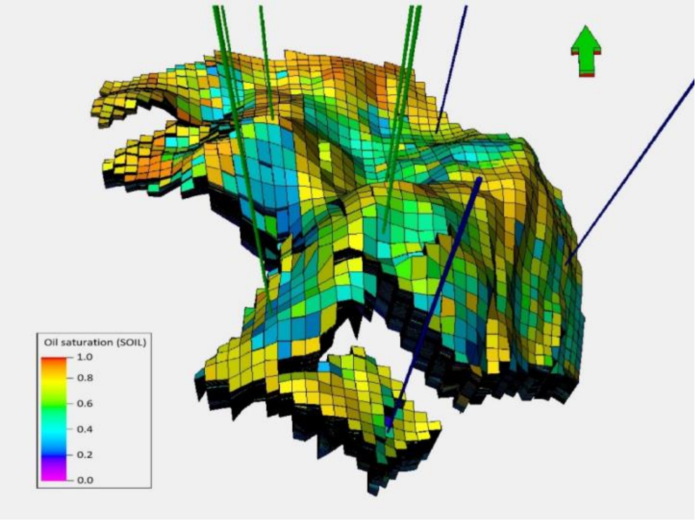


Figure 14. Volve simulation model showing the oil saturation

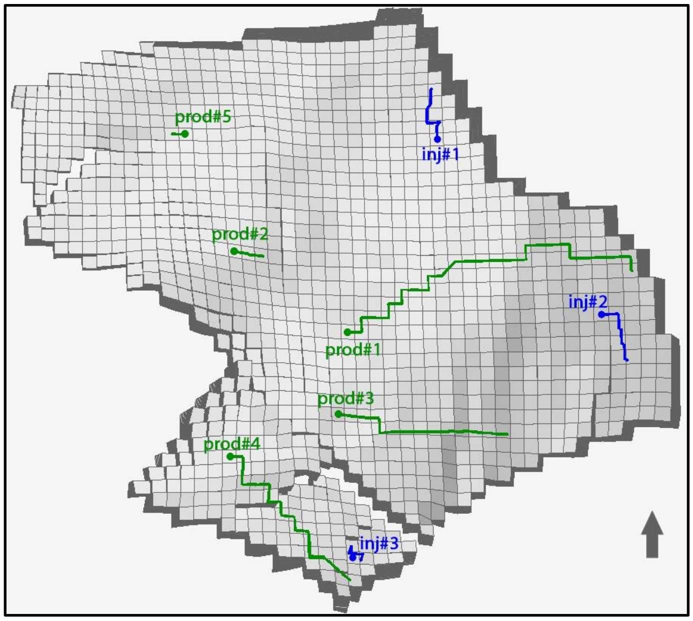


Figure 15. Well locations and their paths in Volve reservoir model

An SPM of the Volve field was constructed to screen the oil saturation and pressure during waterflooding more quickly than could be achieved using the reservoir simulation model. The chosen design parameters were the individual liquid production rate of the five wells. As there are only five design parameters under investigation, a sensitivity analysis was not performed. A genetic algorithm coupled with sequential LHS sampling was used to create the initial sample points, and an optimized algorithm was used to augment more sample points. Then, the simulator software was used to run for ten more years (from 2017 to 2027), and every six months was considered as a timestep (or an output) for the grids' oil saturation and pressure in the new formed dataset. A combination of 23 different models were tested and averaged for feature ranking. A 1D-CNN model was used to train a machine learning model on the data. Finally, the model was validated by evaluating the goodness of fit using RMSE and R^2.

3.4. Results and Discussion

3.4.1. Sequential LHS Design

As explained in the methodology for SPM, the design parameters with their ranges need to be selected. The design parameters are then used in the sampling step to run the model in various configurations. We used the individual liquid production rate (LPR) of the producers as the designed parameters, and a range of zero to 5000 Sm^3/d was chosen. The water injection rates for the individual injectors were not considered as design parameters in this study. In the Volve field, the maximum BHP of injectors is met very quickly even under oil production and low injection rates. As a result, they are not suitable parameters to be chosen for further purposes such as optimization. Table 4 presents the initial sampling points generated based on the LHS design with GA for producers' LPRs.

Table 4. Initial sampling design based on LHS with GA

Sample point	LPR (Sm^3/d) Prod#1	LPR (Sm^3/d) Prod#2	LPR (Sm^3/d) Prod#3	LPR (Sm^3/d) Prod#4	LPR (Sm^3/d) Prod#5	LPR (Sm^3/d) Total
1	68	1,521	2,632	3,665	4,027	11,912
2	2,142	4,936	1,268	4,450	1,942	14,739
3	1,956	2,416	4,764	3,134	323	12,594
4	4,964	3,812	4,282	2,601	3,153	18,812
5	3,336	3,098	882	1,688	4,887	13,892
6	1,434	848	321	2,305	1,386	6,293
7	4,118	1,084	2,001	4,915	2,003	14,121
8	2,812	134	3,894	1,308	3,966	12,115
9	866	4,257	3,473	921	2,850	12,366
10	3,781	2,911	1,502	240	777	9,210

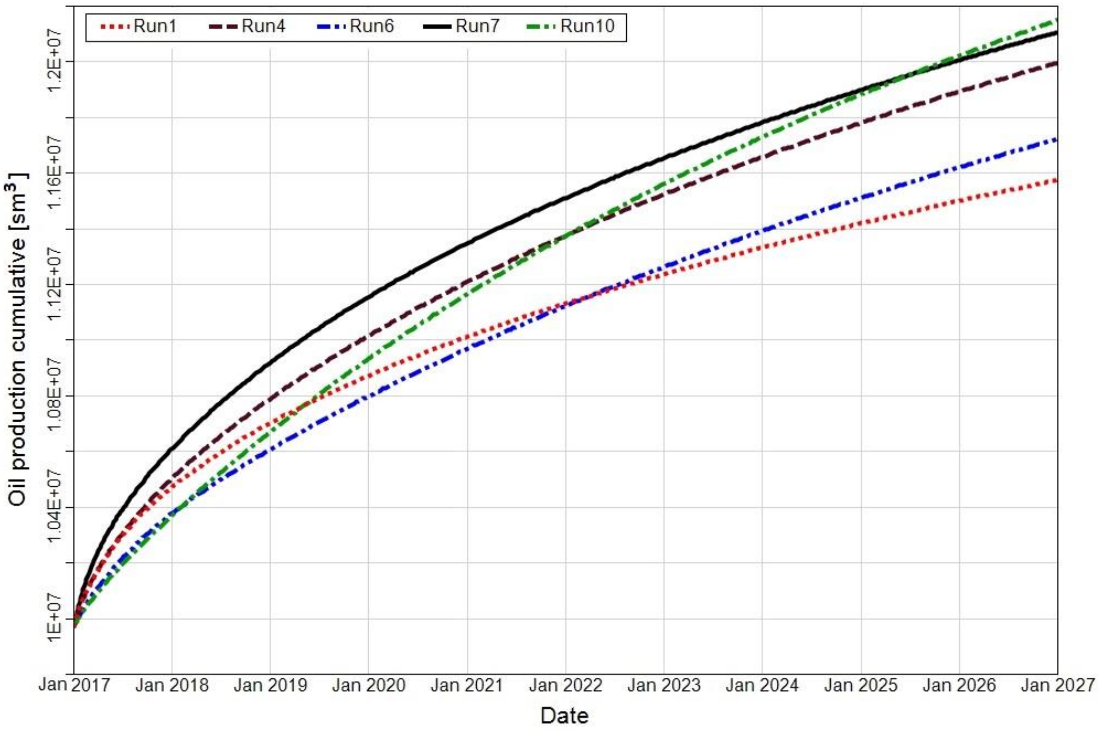


Figure 16. Field cumulative oil production at different sample points

In the initial design, a total of 10 sample points was considered. The simulator software was run on each of the sample points for ten more years (from 2017 to 2027), and the field cumulative oil production for each is shown in Figure 16. By comparing the total LPRs (from Table 4) and the total cumulative produced oil (from Figure 16) it can be concluded that by having a higher total LPR, we can not guarantee that a higher total cumulative produced oil would be achieved over time. Hence, an optimization problem clearly exists when the individual well LPR is selected as the designing parameters that can be used for future research. For example, the maximum total LPR of 18,812 Sm^3/d that corresponds to sample point 4 does not result in the highest total cumulative produced oil after ten years of water flooding. Also, the minimum total LPR of 6,293 Sm^3/d, which corresponds to sample point 6, does not result in the least total cumulative produced oil.

The extracted data from the simulator in these 10 sample points was used to create a dataset and construct the grid-based SPM. Subsequently, five more sample points (shown in Table 5) were added based on the optimized augmentation method to the initial sample points. Adding more sample points to extract more data eventually increased the model accuracy.

Table 5. Added sample points based on the optimized augmentation method

Sample point	LPR (Sm^3/d) Well#1	LPR (Sm^3/d) Well#2	LPR (Sm^3/d) Well#3	LPR (Sm^3/d) Well#4	LPR (Sm^3/d) Well#5	LPR (Sm^3/d) Total
11	364	423	4484	596	412	6,279
12	4,470	4,558	605	4,032	4,589	18,255
13	1,218	3,407	3205	3,697	1,114	12,642
14	3,278	1,970	1981	1,536	3,467	12,231
15	2,653	2,091	2815	2,972	2,458	12,989

In summary, considering a sequential sampling method is crucial because there is no rule to know how many sample points are needed to construct the SPM. This sampling strategy helps extract more data in the domain space of the parameters in the case of low accuracy of the constructed model.

3.4.2. Data Extraction and Dataset Preparation

After running the simulator at the sample points, different static and dynamic data was extracted. Table 6 lists parameters that were used to create the grid-based dataset.

Table 6. Static and dynamic parameters extracted for the grid-based dataset

Parameter	Details
Static	
LHS design parameters	LPRs for producers
Grid indexing	i, j, k
Grid location	X, Y, Z
Grid type	injector, producer, null
Distances of grids	to the closest fault
	to the 1st closest, 2nd closest, and 3rd closest injectors
	to the 1st closest, 2nd closest, and 3rd closest producers
Rock properties	porosity, permX, permY, permZ, TransX, TransY, TransZ, NTG
Grid geometry	vertical thickness, stratigraphic thickness, volume, angle
Grid initial properties	So(t), P(t)
Dynamic	
Grid oil saturation	So(t+1), So(t+2), …
Grid pressure	P(t+1), P(t+2), …

In the formed dataset for the grid-based model, the number of columns equals the number of parameters, and the number of rows is the multiplication of grid numbers and the run numbers (is also equal to the sample points). It means, in this case, that the model consists of 680,400 grids; the number of rows is 680,400×15 with the initial design. The configuration of the grid-based dataset is shown in Figure 17. To decrease the size of the dataset, a preprocessing filter was applied to the dataset to remove the corresponding rows of the grids with zero oil saturation. The grids located in the aquifer have zero oil saturation, and they are not of the interest for any predictions. By applying a filter to remove aquifer grids, the number of rows was reduced to 61,416×15.

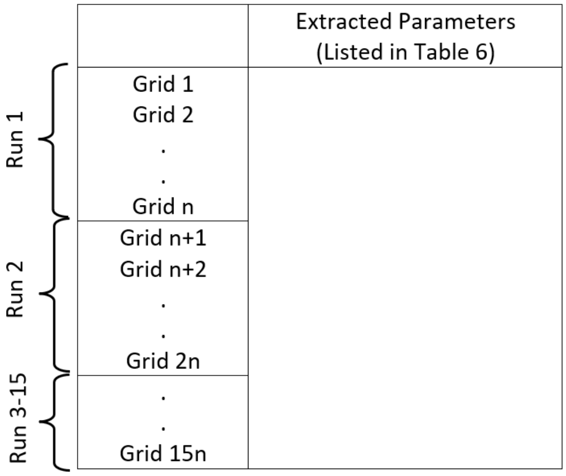


Figure 17. Grid-based dataset configuration

3.4.3. Feature Ranking and Feature Selection

The average feature ranking method using *fscaret* was tested on the grid-based dataset at timestep 7 (Jul 2020). Figures 18 and 19 show the results of feature ranking when the output is the grids' pressure and oil saturation. Figures 20 and 21 also demonstrate the result of feature ranking on the same dataset using the random forest method. The figures clearly show that magnitude and the order of influential parameters are very different between the results obtained from average and random forest methods.

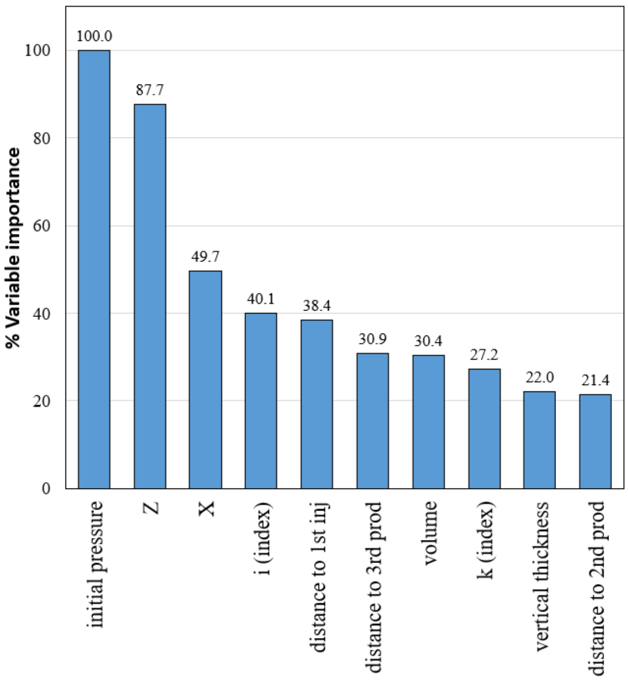


Figure 18. Parameter importance on grids' pressure using average feature ranking method (timestep 7)

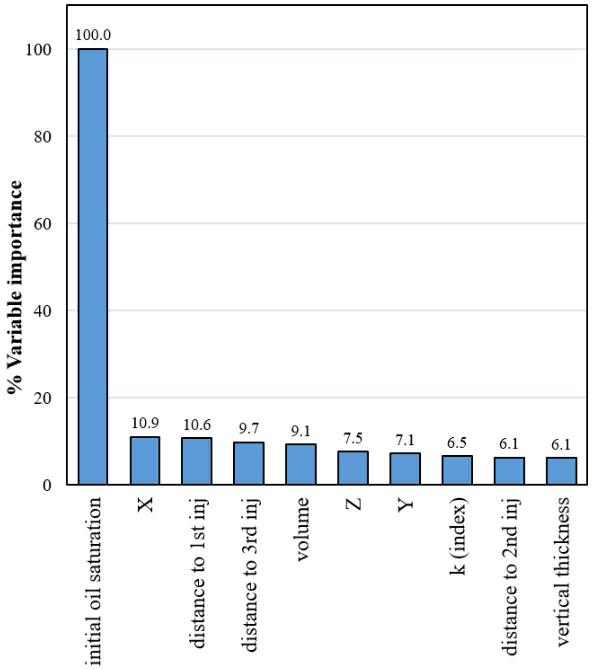


Figure 19. Parameter importance on grids' oil saturation using average feature ranking method (timestep 7)

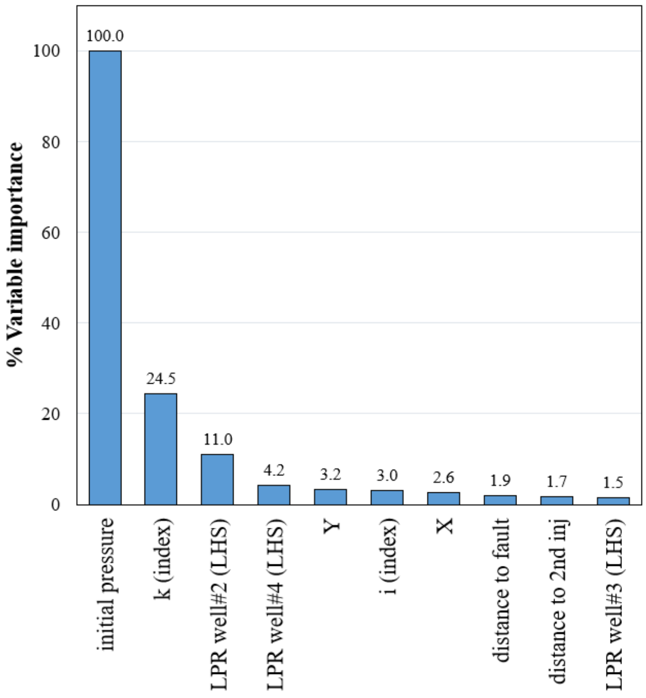


Figure 20. Parameter importance on grids' pressure using random forest feature ranking method (timestep 7)

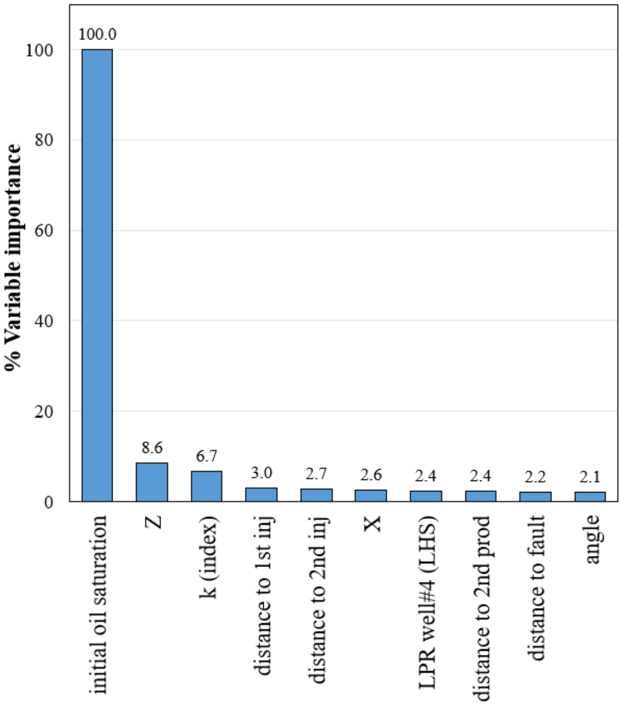


Figure 21. Parameter importance on grids' oil saturation using random forest feature ranking method (timestep 7)

Figure 18 shows that the most important parameter is the initial pressure, and other parameters relating to the grids' location (X and Z) also have a high impact on the grids' pressure in the next timestep (output). However, Figure 19 shows that only the initial oil saturation of the grids has a big impact on the output. This may be due to the fact that a small percentage of recoverable oil is left in the reservoir, and 9.98 million Sm^3 out of 11.4 million Sm^3 expected recoverable oil is produced by 2016. Figures 22 and 23 show the pressure and the oil saturation histograms of the grids. The mean oil saturation around 0.3 from Figure 23 proves this justification.

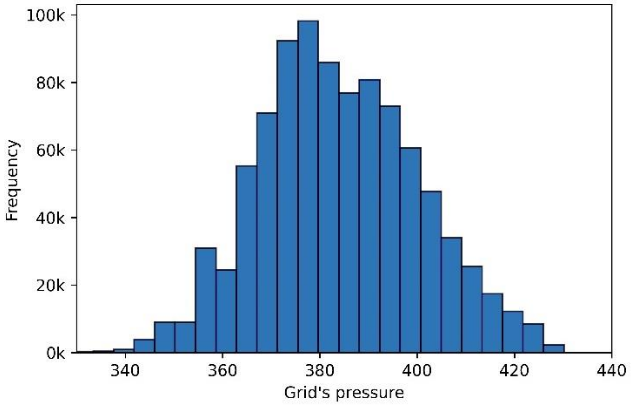


Figure 22. Pressure histogram of the grids

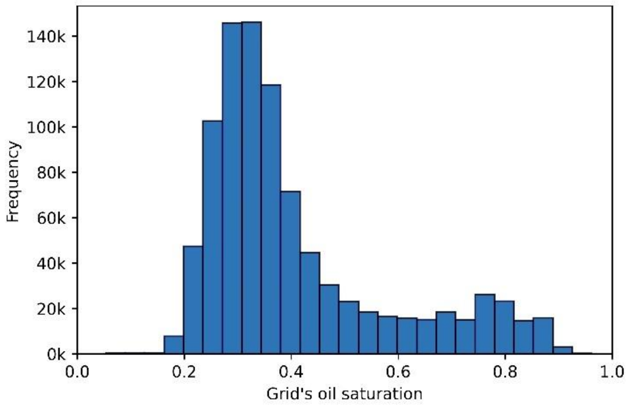


Figure 23. Oil saturation histogram of the grids

Also, another reason for the high impact of the initial saturation might be that the grid-based dataset is representative of all grids in the reservoir model. Nevertheless, only a small percentage of the grids are located near the wells, and injected water or produced liquid could not affect all grids in the reservoir model. To check on this hypothesis, the dataset was investigated in two stages to check the impact of parameters on the grids' oil saturation in the vicinity of wells. In the first stage, the grid-based dataset corresponding to all grids in the reservoir model was shrunk to all grids within a 60 m radius of the injection and production wells. In the second stage, the dataset further shrunk to the corresponding grids in 30 m vicinity of the wells. The feature ranking results for these two stages are presented in Figures 24(a) to 24(d).

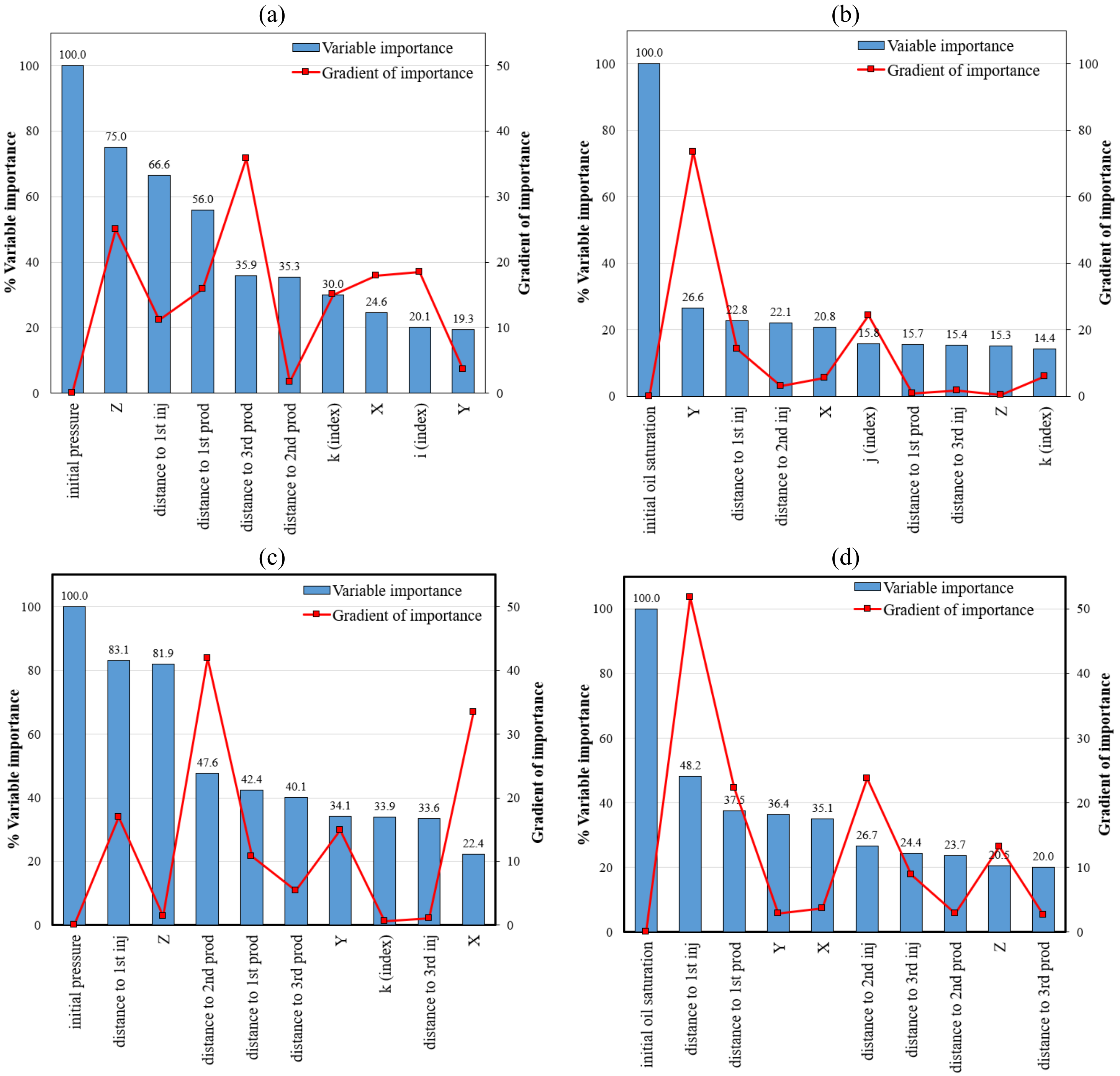


Figure 24. Parameter importance and gradient of importance using average feature ranking method (timestep 7) in vicinity of wells for (a) grids' pressure dataset including grids within 60 m of wells, (b) grids' oil saturation dataset including grids within 60 m of wells, (c) grids' pressure dataset including grids within 30 m of wells, and (d) grids' oil saturation dataset including grids within 30 m of wells

Based on Figures 24(b) and 24(d) (relating to oil saturation), it can be concluded that the more the dataset is limited to grids in the vicinity of wells (tighter radius around the wells), the greater the influence of other parameters. Also, limiting the datasets to a shorter range of wells causes the important parameters to shift toward parameters relating to the production rates, grids' location and distances to wells.

In the feature selection step, different subsets of parameters were chosen with respect to the gradient of importance (for the datasets including grids within 30 m of the wells). Accordingly, wherever the gradient of importance was greater than ten, the number of features prior to that point was selected as a subset. For example, in Figure 24c, the gradient of importance in the 2nd, 4th, 5th, 7th, 10th, and 22nd parameters were all greater than ten, and consequently we selected the subsets containing 1, 3, 4, 6, 9, and 21 parameters. Then, the model accuracy on each subset was checked to select the optimal subset.

3.4.4. Model Training/Validating

The goodness of fit for ANN for different subsets was obtained at the initial effort. Accordingly, 80 percent of the dataset was randomly picked for the training set, and the rest 20 percent was picked for the test set. Furthermore, 10 percent of the training set was selected to do the validation and to ensure that the overfitting does not happen. The ANN models gave the best results with two hidden layers (each containing 200 neurons). The results of the ANN models at different subsets of parameters in timestep seven are presented in Table 7. Accordingly, the subsets containing the first nine parameters (for the pressure dataset) and 16 parameters (for the oil saturation dataset) were chosen for further analysis.

Table 7. ANN models at different subsets of parameters in timestep 7

Pressure			Oil saturation		
No. features in different subsets	R^2 (test set)	RMSE (test set)	No. features in different subsets	R^2 (test set)	RMSE (test set)
1	61.25	80.7623	1	55.39	0.0111
3	78.79	45.8134	2	66.71	0.0084
4	83.29	35.0527	5	75.22	0.0063
6	88.42	25.4004	8	82.62	0.0045
9	91.50	18.9462	16	87.39	0.0033
21	92.13	17.0063	24	88.28	0.003
31 (all)	92.02	17.5116	31 (all)	89.06	0.0028

In the next effort, we used CNN as a deep learning technique to build more accurate underlying models for the selected subsets. Initially, we focused on optimizing the adjustable hyper-parameters of the CNN model for the selected datasets. Table 8 demonstrates the selected adjusting hyper-parameters with their ranges to be optimized. D-optimality design of the experiment was applied to the grid-based datasets. By analyzing the results of the D-optimality test, the optimum value for the CNN adjusting parameters was achieved, as shown in Table 9. These values for the CNN parameters were then applied to each dataset and different timesteps. Table 10 presents examples of the trained CNN models corresponding to the datasets resulting from 10 sample points.

Table 8. Adjusting parameters for CNN model and their ranges

Parameter	Range
Filter size 1	10 - 90
Filter size 2	10 - 90
Filter size 3	10 - 90
Kernel size	2 - 3
Neuron size 1	10 - 90
Neuron size 2	10 - 90

Table 9. Optimum values for parameters resulting from D-optimality test

Parameter	Value
Filter size 1	90
Filter size 2	85
Filter size 3	78
Kernel size	3
Neuron size 1	60
Neuron size 2	90

Table 10. R^2 and RMSE of CNN models for datasets corresponding to 10 sample points at different timesteps

Date	Timestep	Pressure		Oil saturation	
		R^2 (test set)	RMSE (test set)	R^2 (test set)	RMSE (test set)
JUL 2017	1	98.88	1.9137	94.91	0.00145
JUL 2020	7	98.72	2.1528	94.77	0.00149
JUL 2023	13	98.76	2.0803	94.36	0.00160
JUL 2026	19	98.61	2.3584	94.30	0.00161

Comparing the results obtained from CNN and ANN models, we see that the CNN models worked better than the ANN models. To check if the trained models work well on unseen data, a random blind test based on the designed parameters of the SPM was introduced (Table 11), and the accuracy of the CNN models was checked. Table 12 presents the results of the CNN model for both grids' oil saturation and pressure in the datasets resulted from ten sample points for this blind test.

Table 11. Blind test for waterflooding scenario

	LPR well #1	LPR well #2	LPR well #3	LPR well #4	LPR well #5
Blind Test	4,399	2,605	308	4,149	4,782

Table 12. R^2 and RMSE of CNN models for blind test, and for datasets corresponding to 10 sample points

Date	Timestep	Pressure		Oil saturation	
		R^2	RMSE	R^2	RMSE
JUL 2017	1	81.63	38.4685	79.43	0.00525
JUL 2020	7	81.69	38.3693	79.39	0.00526
JUL 2023	13	81.57	38.6394	79.22	0.00529
JUL 2026	19	81.41	38.9835	79.36	0.00526

The results show that, unlike the good accuracy of the trained models around 94- 99%, the CNN models did not give promising results for the blind test. Hence, as discussed in Section 3.4.1, five more sample points were added to the ten initial sample points using sequential sampling to cover more domain space for the design parameters. The CNN model was constructed again for the new datasets at different timesteps. Table 13 presents the accuracy of the constructed CNN models in some of the timesteps as examples. This time the results obtained from the blind test showed an improved accuracy of around 92-96%, as shown in Table 14. Consequently, these SPMs can reliably be applied for screening purposes.

Table 13. R^2 and RMSE of CNN models for datasets corresponding 15 sample points at different timesteps

Date	Timestep	Pressure		Oil saturation	
		R^2 (test set)	RMSE (test set)	R^2 (test set)	RMSE (test set)
JUL 2017	1	98.89	2.3682	94.17	0.00167
JUL 2020	7	98.83	2.4826	93.93	0.00173
JUL 2023	13	98.61	3.0374	93.86	0.00175
JUL 2026	19	98.55	3.2262	93.91	0.00174

Table 14. R^2 and RMSE of CNN models for blind test, and for datasets corresponding to 15 sample points

Date	Timestep	Pressure		Oil saturation	
		R^2	RMSE	R^2	RMSE
JUL 2017	1	96.18	7.6266	92.63	0.00203
JUL 2020	7	96.16	7.7230	92.59	0.00205
JUL 2023	13	95.87	8.7157	92.38	0.00211
JUL 2026	19	95.77	9.1406	92.41	0.00210

In the last effort, to prove that the SPM outperforms the TPM, we excluded all the extracted parameters from the datasets. This dataset only included the design parameters (LPRs for individual wells) and the output. However, such a dataset cannot be used to predict the grids' oil saturation and pressure, and the training models overfit.

3.4.5. Screening

The target parameters for screening in this study were the oil saturation and pressure of the grid blocks. The results obtained from the blind test at layer one and different timesteps are presented in Figures 25 (grid oil saturation) and 26 (grid pressure). In these figures, the predicted values from the SPM, the predicted values from the simulator (i.e., actual values), and the error between them are demonstrated. More figures related to other layers of the Volve model can be found in Appendix A. From the oil saturation figures, the effect of water injection and oil production can be seen. The figures can be used for further decision-making in the field. The same figures for different layers, timesteps, and blind production rates of the wells can be generated by running the SPM, and it only takes a few seconds to screen the result while using a simulator might take hours for a single run. To be more specific, each simulator run for the Volve case took around five hours on a machine with 16 GB RAM, and 3.20 GHz processor. This run time was only four seconds for each SPM deployment on the same machine. This difference between using the numerical model and the SPM will be even more significant if we desire to have more runs. For example, 100 runs for the simulator will take three weeks, but this number would be around seven minutes for SPM, which is very considerable. It should be mentioned that Volve model is a small case that only includes five production wells. The difference can be substantially larger for a bigger reservoir model with more complexities. In the work presented by Gholami et al. [8], they compared SPM and black oil simulation time for the SACROC field. They concluded that the simulator would take seven months and six years to perform 100 and 1000 runs, respectively.

It is worth mentioning that the process of constructing the SPM, including the research, numerical model tunning, coding the full workflow, getting results, and validating the whole procedure, might be tedious and time-consuming for the first time. It took us around two years for the first case including learning what SPM models were and literature review. However, the second case only took two weeks as the process was well organized in Python code. Most of this time was also spent on running the numerical model at the sample points and training the underlying models.

Finally, to quality check the CNN results, the percentage error histogram for both the oil saturation and pressure datasets in the blind test were achieved (Figure 27). The mean value is very close to zero, and the high frequency in ranges between -0.05% and 0.05% again shows the accurate results obtained from the CNN model and the consequent SPM.

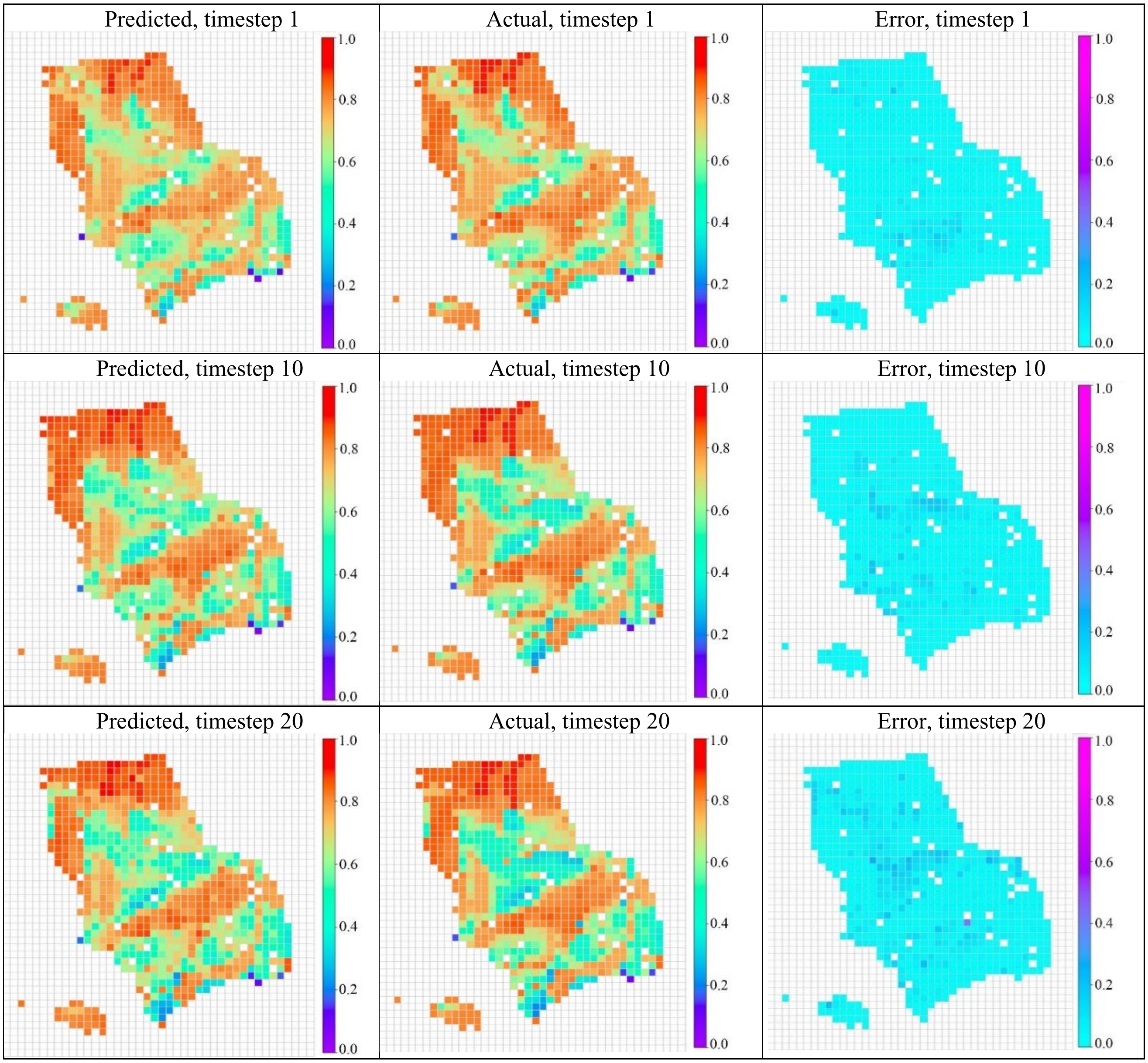


Figure 25. Grids oil saturation in layer 1 for the blind test (waterflooding case)

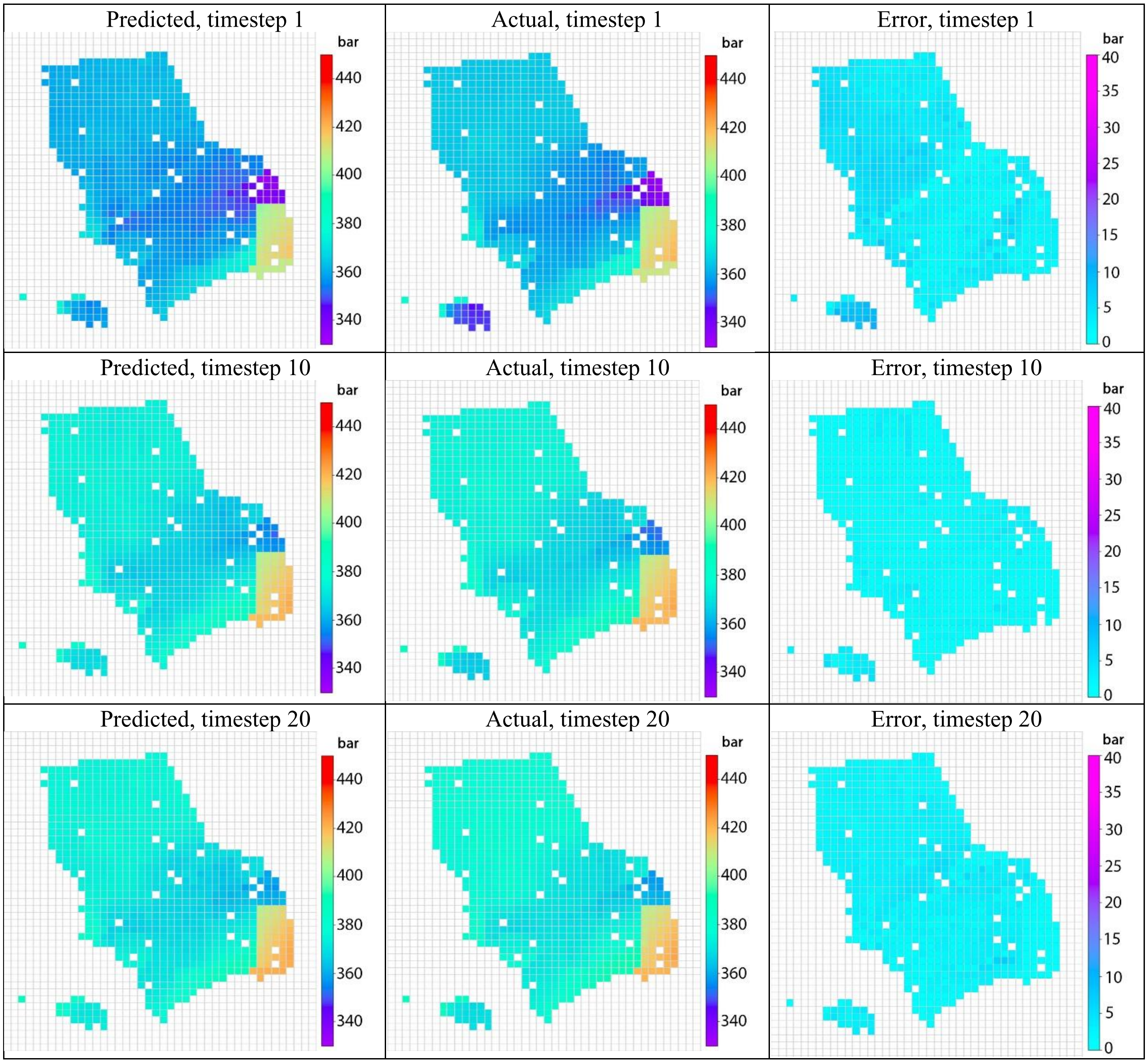


Figure 26. Grids pressure in layer 1 for the blind test (waterflooding case)

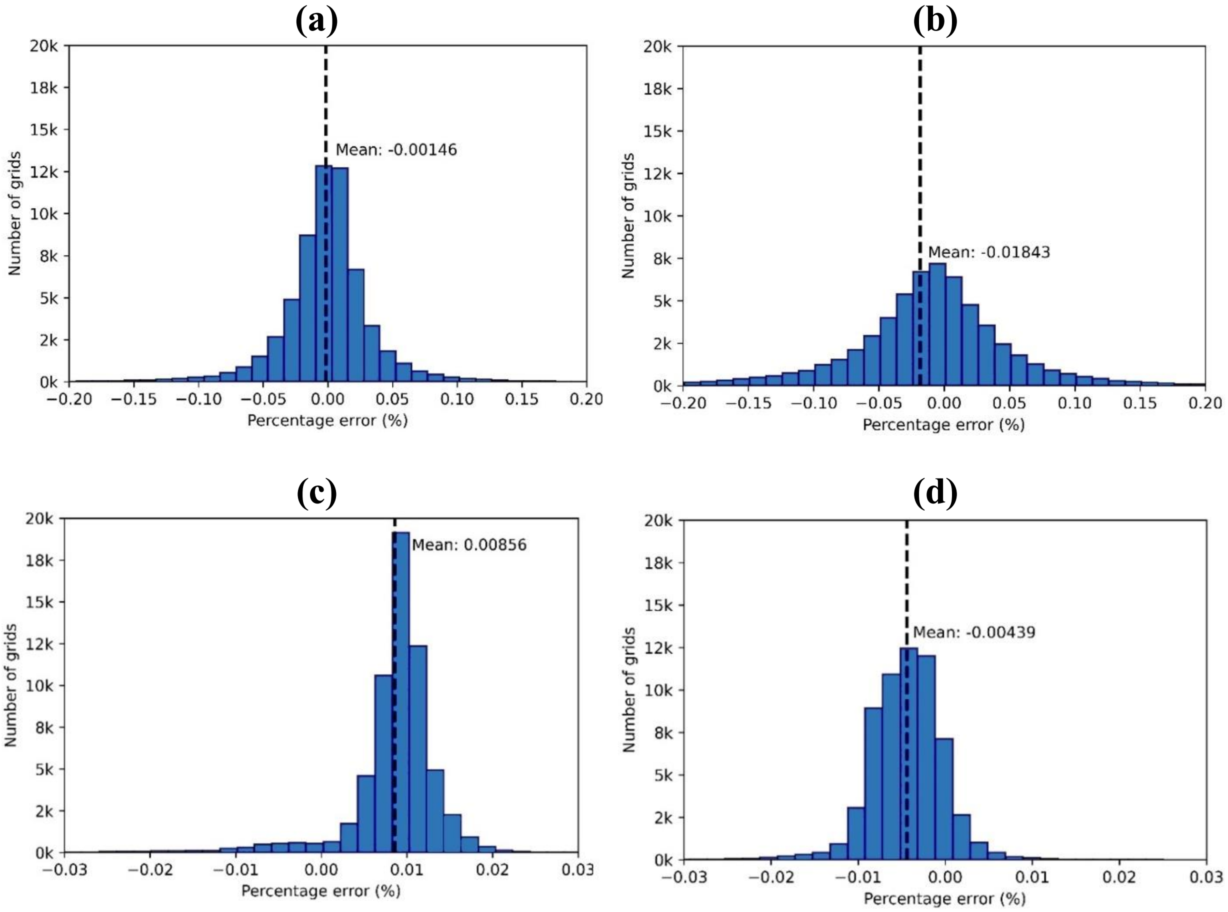


Figure 27. Percentage error histogram for the datasets corresponding to (a) oil saturation at timestep 1, (b) oil saturation at timestep 20, (c) pressure at timestep 1, and (d) pressure at timestep 20

3.5. Conclusions

The objective of this paper was to investigate accuracy and efficiency improvements to smart proxy modeling (SPM) introduced by Mohaghegh [7]. We used a grid-based the SPM workflow for a waterflooding using the Volve field as a case study. Based on our literature review, we specifically targeted our investigation into techniques in sampling (sequential sampling), feature ranking, and underlying model construction (CNN instead of ANN).

Sampling consisted of choosing individual liquid production rates as our design parameters where a novel sequential sampling technique (LHS with a GA) was applied. Initially, ten sample points were introduced to run the numerical model, but five more sample points were later added to achieve the highest accuracy in the lowest number of the runs. The sequential sampling used in this work helped to construct the final SPM more efficiently rather than resampling the whole procedure used in the literature. After running the simulator, different static and dynamic parameters were extracted for all individual grids. The extracted data for the grids was used to form a new dataset called a grid-based dataset. In this work, a new feature ranking method is proposed, which is an average of 23 models. The average feature ranking technique described in this work provides a more confident prioritization of input parameters compared to the case of considering only one model which eventually helped the overall efficiency in the feature selection step. Additionally, feature ranking results showed that some of the static and dynamic parameters extracted from the simulator are among the most influential parameters. Next, a one-dimensional CNN model (as a novel deep learning technique) was used to predict the grids' oil saturation and pressure. The goodness of fit for both ANN and CNN models were tested. The SPM with ANN underlying model provided an accuracy of 89-92% compared to the 99% and 94% of the CNN technique for the pressure and oil saturation predictions, respectively. Also, the accuracy of the SPM used in this work was tested for different timesteps and showed an accuracy of 92-96% for the blind test.

The constructed SPM in this work was capable of reproducing the results of the simulator at very high accuracy in a very short time. The SPM can be used for screening various EOR scenarios and any parameters at the grid level, such as gas saturation, component concentration, and many others.

Chapter 4. Field Production Optimization Using Smart Proxy Modeling; Implementation of Sequential Sampling, Average Feature Ranking, and Convolutional Neural Network

Preface

A version of this chapter has been accepted at the 23 SPE Canadian Energy Technology Conference and Exhibition. I am the primary author, along with the co-author Dr. Lesley A. James. The detail of the author's contributions are as follow:

Peyman Bahrami: Conceptualization, Methodology, Writing–original draft, Writing–Review & Editing, Visualization

Dr. Lesley A. James: Conceptualization, Methodology, Writing–Review & Editing, and Supervision

Reference: P. Bahrami and L. A. James, "Field Production Optimization Using Smart Proxy Modeling; Implementation of Sequential Sampling, Average Feature Ranking, and Convolutional Neural Network", Presented at 23 SPE Canadian Energy Technology Conference and Exhibition, Calgary, Canada, March 2023, SPE-212809-MS, https://doi.org/10.2118/212809-MS

Abstract

This work aims to create an approximation of the reservoir numerical model using smart proxy modeling (SPM) to be used for production optimization. The constructed SPM in this work is further improved in different steps to increase its accuracy and efficiency compared to the existing literature. These steps include sequential sampling, average feature ranking, convolutional neural network (CNN) deep learning modeling, and feature engineering.

SPM is a novel methodology that generates results faster than numerical models. SPM decouples the mathematical equations of the problem into a numeric dataset and trains a statistical/AI-driven model on the dataset. Major SPM construction steps are: objective, input, and output selection, sampling, running numerical model, extracting new static and dynamic parameters, forming a new dataset, performing feature selection, training and validating the underlying model, and employing the SPM. Unlike traditional proxy modeling, SPM implements feature engineering techniques that generate new static/dynamic parameters. The extracted parameters help to capture hidden patterns within the dataset, eventually increasing SPMs' accuracy.

SPM can either be constructed to predict the grids' characteristics, called grid-based SPM, or to predict the wells' fluid rates, called well-based SPM. In this work, the well-based SPM is constructed to duplicate the Volve offshore field production results undergoing waterflooding. We used Latin hypercube sampling coupled with genetic algorithm (GA) in the sampling step. The designed parameters to perform sampling are the individual liquid rate of the producers, and the output is the individual well's cumulative oil production. In the formed dataset, various extracted parameters relating to the wells are prepared, such as well types, indexes, trajectories, and cumulative oil production. Furthermore, a grid-based SPM is constructed in parallel to the well-based SPM. At each timestep of the prediction, dynamic parameters relating to grids (in this case: grids' pressure/saturations) are transferred to the existing well-based dataset. This technique helps the well-based SPM further increase in accuracy by finding new patterns within the dataset. We implement an average of 23 different models to rank, and perform the feature selection process. Finally, the CNN model is trained on the dataset, and is coupled with two derivative-free optimizers of GA and particle swarm optimizer to maximize the oil production over the selected time period.

Sequential sampling used in this work is a novel technique to construct the SPM with the lowest number of numerical model executions. It provides an efficient workflow to perform sampling, thereby saving time instead of repeating the whole SPM construction steps. The average feature ranking implemented in this paper provides the best prioritization of input parameters. It provides a confident ranking for the feature selection step. Finally, the underlying CNN model is compared to the prediction accuracy of the ANN model.

Keywords: Smart proxy model; Machine learning; Convolutional neural network; Sequential sampling; Average feature ranking

4.1. Introduction

During field development planning and production, the reservoir model is considered one of the main tools to predict fluid production. Reservoir models are categorized as either static or dynamic. A static reservoir model includes geological parameters related to the architecture of the reservoir, such as the lithologies, porous media characteristics, and barriers [224]. These parameters remain almost unchanged in the lifetime of the reservoir during production and are known as static parameters. A conventional static model usually consists of millions of grid blocks for geological characterization, which may increase to hundreds of millions in more complex reservoirs [225]. A dynamic reservoir model is about the movements of fluids within the porous media as a function of the pressure gradient [226]. Since the wells produce the fluids, the parameters related to fluid movement and pressure vary over time and are recognized as dynamic parameters. A numerical reservoir model incorporates static and dynamic parameters and implements partial differential equations to characterize the fluid pressure and saturations in grid blocks over time [227]. Engineers tend to reduce the size of grids as much as possible to achieve more accurate predictions and capture the complexities within the reservoir using numerical models. Smaller grid size means having more grids and requiring higher computation power to solve all partial differential equations. In recent years, even with improvements in computation power such as parallel computing, users desire to have higher quality results with less simplifying assumptions and smaller grids. As a result, the need for computer resources increases exponentially, hence researchers try to implement alternative methods to speed up the process of prediction from the numerical models.

While one single run for a numerical model may take hours to perform, optimizing field development plans that require hundreds of runs takes days or months to be completed. Proxy modeling (PM) is an alternative method to reproduce the results from numerical models, with run times in seconds rather than days or months. PMs are also referred to as surrogate models, meta models, and RSM in the literature. Since the output from a numerical model provides the data to train the PM, an accurate numerical model is the main requirement to construct a PM. PMs can be classified into traditional proxy models (TPM) and smart proxy models (SPM) based on development strategy [215]. TPM usually implements statistical/AI-driven models and sometimes can use simplified theories in physics or dimensional reduction approaches to relate the input parameters to the output parameter. The methodology of constructing a TPM includes running the numerical models, recording the outputs, and making a relationship between the inputs and outputs. TPMs usually require a higher number of numerical model runs and suffer in terms of prediction accuracy [9]. As with traditional models, SPM also creates a methodology to relate inputs to outputs. However, SPMs work based on pattern recognition and machine learning methods and have additional steps in the construction process compared to TPMs. The main additional steps in SPM construction are extracting new static/dynamic parameters and forming a new dataset. The newly extracted parameters help to capture unseen patterns within the reservoir under investigation.

In this work, an SPM is developed to mimic the individual well's cumulative oil production during a waterflooding process. Even though waterflooding is one of the most widely used secondary recovery methods, finding an optimal selection of the design parameters is challenging. Selecting suitable values for the controlling parameters of the injectors or producers often happens through trial and error, or a design of experiments that involves running a large number of runs. An SPM can alleviate this drawback in waterflood production optimization by providing an alternative approach that, once built, is faster than the numerical models and is more accurate than the TPMs.

This study aims to increase SPM accuracy and efficiency by implementing sequential sampling, feature ranking using the average of multiple algorithms, underlying model selection, and introduction of an applied tiering system for the grids. Latin hypercube sampling (LHS) coupled with a genetic algorithm (GA) is used for the initial sampling design at the lowest number of design points. Then, sequential sampling is used to add more design points to the existing design. Sequential sampling prohibits starting the whole sampling procedure and running the numerical model from the beginning. This saves a lot of time in the construction procedure of the SPM. In the feature ranking step, an average of different algorithms' results is used to make the outcome of this step more reliable.

Furthermore, the implemented tiering system helps to feature engineer and generate a new set of parameters. These new parameters make it possible to reveal new patterns within the formed dataset and eventually increase the accuracy of the SPM. The process of forming the tiering system and extracting the parameters happens in a hybrid implementation of two different SPMs, which will be explained in detail in the dataset preparation section. SPM implements training an underlying machine learning method. In this work, two algorithms, artificial neural network (ANN) and one dimensional convolutional neural network (1D-CNN) are trained and compared in terms of goodness of fit and accuracy. The validated SPM is then coupled with two optimizers to find the maximum cumulative oil production for the field in an interval of ten years.

4.2. Work Background

As previously discussed, PMs can be classified as TPM and SPM based on the development strategy. TPMs have a more extended history of implementation in the literature and can be constructed through different approaches such as statistical/AI-driven models, simplifying physics, or dimensional reduction concepts. TPMs based on statistical/AI-driven models were used in various reservoir applications such as production optimization [2], history matching [4], uncertainty quantification [3], gas lift optimization [114], waterflooding optimization [185], well placement optimization [188], well control optimization [191], fractured reservoirs screening [180], chemical flooding [196], gas assisted gravity drainage [193], and water alternating gas (WAG) [195]. TPMs based on simplifying physics create an easier model through simplifying assumptions [32] or coarser discretization [31] compared to the high-fidelity model, and it is usually recognized as multi-fidelity proxy models in the literature. There are some applications of this category in petroleum engineering, such as history matching [133], [139], well placement optimization [135], production optimization [129], and production forecasting [147]. The last category for TPMs is based on the dimensional reduction of the problem in which irrelevant parameters are neglected. This type of TPM is also known as the reduced-order model. Applications of this category include water flooding optimization [152], history matching [154],

[157], fluid flow in porous media [164], and thermal recovery process [159]. The readers are encouraged to refer to the review by Bahrami et al. (2022) to gain more insight into TPM’s categories with more comprehensive applications in reservoir.

There are two classes of SPM: grid-based and well-based. The constructed SPM is called grid-based if it is used to predict the dynamic parameters of the grids, such as pressure and fluid saturations. It is called well-based if the outputs are related to the wells, such as fluid productions. There are limited applications of SPM in the literature. Grid-based SPM is applied to waterflood monitoring [10], [197], gas injection monitoring [11], WAG monitoring [8], and CO_2 storage observation [228]. Examples of well-based SPM are in history matching [9], [12], and production optimization [8].

4.3. SPM Development Workflow

The workflow for the SPM development is presented in Figure 28. The initial step in any proxy modeling development is objective determination. The objective might be production optimization, uncertainty analysis, history matching, etc. After setting the objective, different design parameters and their ranges are selected. The objective of our case study is production optimization over a fixed period of time in waterflooding. The design parameters are the individual liquid production rate (LPR) of the wells.

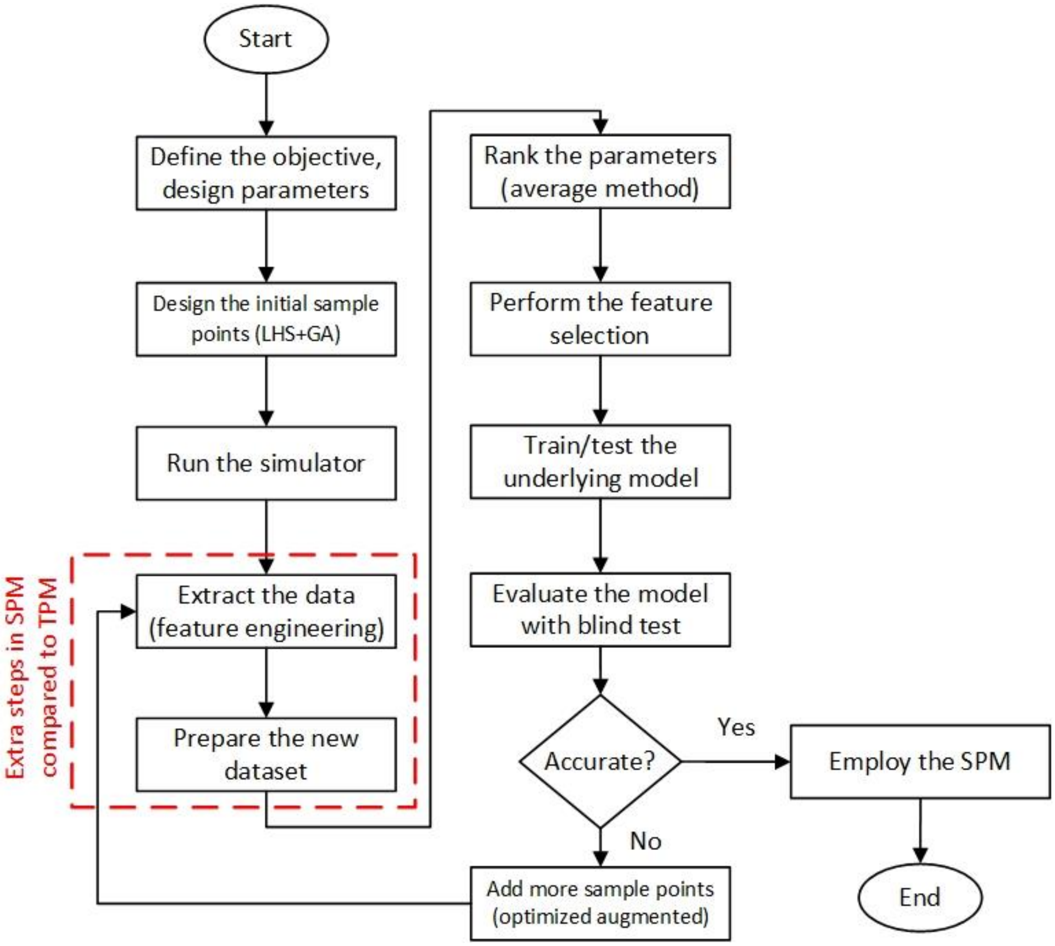

Start
Define the objective, design parameters
Design the initial sample points (LHS+GA)
Run the simulator
Extract the data (feature engineering)
Prepare the new dataset
Extra steps in SPM compared to TPM
Rank the parameters (average method)
Perform the feature selection
Train/test the underlying model
Evaluate the model with blind test
Accurate?
Yes
No
Employ the SPM
End
Add more sample points (optimized augmented)

Figure 28. Workflow for the SPM methodology with all essential steps performed in this work

The next main step is sampling. Sampling is the process of using the design of experiment methods to select a set of design or sample points over the domain space of the designing parameters. Then, the numerical model runs at each sample point to extract the desired data. Since the main reason to construct an SPM is to speed-up reduplicating the numerical model outputs, it is vital to construct the SPM with the least number of sample points, making sampling very challenging. If we waste so much time running the numerical model at a high number of sample points, the applicability of SPM diminishes. On the other hand, if the number of sample points is low, the constructed SPM may not capture the main patterns within the problem, and we lose the prediction accuracy. There is no scientific procedure to choose the optimum number of sample points for a given problem. The sample size depends on the heterogeneity of the problem under investigation.

To solve this challenge, we use sequential sampling. Based on a literature review, this sampling approach has not previously been used for SPM construction. Unlike stationary sampling techniques that have a fixed number of sample points, in sequential sampling, sample points can be added until the desired coverage of the domain is achieved and the performance is acceptable [72]. In this work, LHS is the main algorithm for the initial design. It is possible to start the LHS design with any number of sample points, and since the goal is to construct the SPM with the lowest number of sample points, this is a significant advantage. Other sampling techniques, such as factorial and optimal designs, have a fixed structure for design, and it is impossible to set an optional number of sample points in the domain space of the parameters. SPM uses pattern recognition techniques by extracting extra parameters from the numerical model runs, and it can provide more insight into the problem under investigation with fewer runs. That is why the LHS needs to be implemented in SPM construction to control the number of sample points. LHS is considered a stratified approach that divides each designing parameter into equal intervals depending on the sample point size. Then, it distributes the sample points in each interval of the parameters. However, the LHS can still not be considered a perfect algorithm, and there are chances that the algorithm distributes the sample points non-uniformly. So, it is important to optimize the distance between sample points.

In this work, LHS is coupled with GA, which works based on the S-optimality criterion. S-optimality looks for the maximum mean distances between sample points [229]. It is possible to add more sample points to the initial design in sequential sampling. This is also done by implementing an optimized algorithm based on the S-optimality criterion, just like the initial design.

After running the numerical model at the sample points, different static and dynamic parameters are extracted from each run and stored in a new dataset. This process is called feature engineering. The new dataset includes the initially selected designing parameters for the objective, the new extracted parameters, and the outputs. The initial designing parameters of this work are the individual LPR for the wells, and the output is the cumulative oil production. The extracted parameters include various parameters such as porosities, permeabilities, saturations, distances, indexes, etc. We introduced a tiering system to consider the impact of the surrounding grids of the wells, and the static and dynamic parameters related to the tiers were imported into the dataset. The rows in the well-based and grid-based datasets correspond to the individual well and grid in the numerical model, respectively. The number of columns equals the number of parameters that are considered for either the well-based or grid-based dataset. Such datasets for grid-based SPM, or well-based SPM (with a high number of wells) would be massive, and reducing the number of rows and parameters before training any underlying model is a must. The number of rows can be reduced by picking a random fraction of all rows, or it is possible to lump the grids before any numerical model executions. For example, in work done by Gholami et al. [8], the authors reduced the number of rows (from 396,000 to 55,000 rows) and the number of columns (1,000 features to 310) through grid lumping in the Z-direction and feature selection techniques. Feature selection may even lower the chance of overfitting during underlying model training. However, the result of feature selection highly depends on the feature ranking step that happens prior to that.

Many predictive models are available to rank the importance of input parameters on the output. These predictive models may result in entirely different ordering and magnitudes for the input parameters and, consequently, change the results of the feature selection step [220]. The *fscaret* package presented by Szlęk et al. [221] provides an ensemble of feature importance based on multiple predictive models. Using the ensemble of algorithms can lead to better-selected parameters and more consistent results [230]. The generated importances are then averaged for each parameter and scaled up to 100 to rank the parameters. Furthermore, *fscaret* provides a list of the gradient of parameters' importance that can be used for the feature selection step. We can search for the optimal subset of parameters based on the gradient of importance. This procedure is similar to the wrapper feature selection techniques that look into different subsets of the parameters.

The underlying model construction involves three main steps of training, testing, and blind test verification. Initially, the dataset splits into training and test sets. Two different models of ANN and 1D-CNN are applied to the training set. Then, the test set that is not used in the training process is applied to the models to check the goodness of fit and prediction accuracy. The goodness of fit is determined in two terms root mean squared error (RMSE) and coefficient of determination (R^2), and the prediction accuracy is obtained by calculating the mean absolute percentage error (MAPE) using Equation 14.

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_{act,i} - y_{prd,i}}{y_{act,i}}\right| \tag{14}$$

where y_{act} is the actual value, y_{prd} is the predicted value, i is the data record number, and n is the total number of records. The ANN model used in this work consists of an input layer, two hidden layers, and an output layer in which the number of hidden layers is chosen by trial and error to give the best prediction accuracy. A 1D-CNN model, which is a type of deep learning technique, has additional layers compared to ANN. These layers are convolutional and pooling layers. The convolutional layer reduces the size of the initial dataset. In this layer, a mathematical calculation (dot product) is applied between the input dataset and a sliding filter. The kernel size for this moving filter is M×N, where M is equal to the height of the dataset, and N is the width of the filter. The convolutional layer can have multiple sub-layers with a different number of filters at each sub-layer. The number of sub-layers and filters can be optimized through trial and error, and it depends on the amount of data and the complexity. The pooling layer decreases the dimensionality and can find the influential parameters. This layer is not implemented in this work as feature selection is already applied to the dataset. The outputs of the pooling layer are flattened and fed into a fully connected layer similar to the ANN model.

Once the underlying model is trained and validated, the SPM can be applied for other tasks such as forecasting or optimization. It is used SPM to optimize the field's cumulative oil production over ten years of production. Optimizers used in this work are particle swarm optimization (PSO) and GA.

PSO is a stochastic optimization algorithm that was first introduced by Kennedy and Eberhart [117] in 1995. PSO is inspired by the social cooperation of animals, such as birds when looking for food. The algorithm determines the velocity and position information of the individual models, which is known as a particle. The group of models forms a swarm, which is a group of solutions. The main objective of the PSO algorithm is to find the global optimum in the domain space of the parameters. Hence, an individual particle updates at each step based on the individual and global best solutions at the previous step.

In PSO, the position of each particle in the search space updates in iterations as follows:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{15}$$

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (P_{i,best} - x_i^t) + c_2 r_2 (G_{best} - x_i^t) \tag{16}$$

where x_i^t is the position vector of particle i at iteration t, and v_i^{t+1} is the velocity vector of particle i at the next iteration, $P_{i,best}$ is the individual best position of particle i from initialization through time t, G_{best} is the swarm's best position from initialization through time t, ω is the inertia weight, c_1 and c_2 are the acceleration coeffiecients, and r_1 and r_2 are the random values between zero and one. Variables c_1, c_2, r_1, and r_2 are usually constant values that are obtained empirically. Term $c_1 r_1 (P_{i,best} - x_i^t)$ is recognized as the cognitive learning component. This term checks the personal history of the particle's position and analyzes the particle’s performance through time. Basically, this term helps each particle to move toward its best personal position. Term $c_2 r_2 (G_{best} - x_i^t)$ is known as the social learning component, which looks into the performance of the particles compared to a group of neighbor particles. The term inertia weight in equation 16 can be considered either fixed or a varying value through iterations. Inertia helps to keep the influence of the previous velocity in the ongoing iteration. The process of calculations for particle positions continues until the particles find the global optimum.

GA is also a stochastic algorithm for optimization introduced by Holland [115] in 1992. GA uses the principles of genetic and natural selection inspired by Darwin’s theory. The algorithm initiates with a population of solutions or chromosomes. Each solution is a set of optimizing parameters and is often expressed as a string of binaries. At each iteration (generation), the fitness values for the solutions are calculated, and the best solutions pass to the next iteration. The algorithm undergoes three main processes of selection, mutation, and crossover at each iteration, and it moves from one generation (parents) to the next (children). Selection is the process of choosing the solutions for breeding in the next generation. Mutation helps to maintain the diversity of solutions from one step to another. On the other hand, crossover recombines the solutions in the previous step to inherit the crucial information for the next step.

PSO and GA are recognized as strong algorithms for finding the global optimum. GA can be applied to continuous and discrete optimization problems, but it is an expensive and time-consuming method if the number of parameters increases. On the contrary, PSO is computationally cheap but is less applicable when the number of parameters increases [116].

4.4. Case Study

A reservoir model of the Volve field is used as the base numerical model to construct the SPM. It is an offshore sandstone reservoir located west of Stavanger, Norway. Figure 29 shows the Volve field reservoir model, and Table 15 presents the key characteristics of the field.

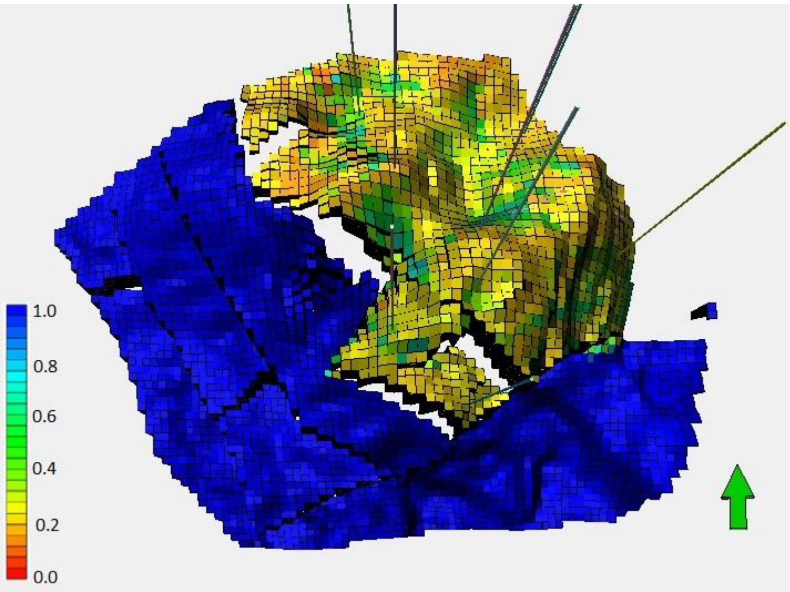
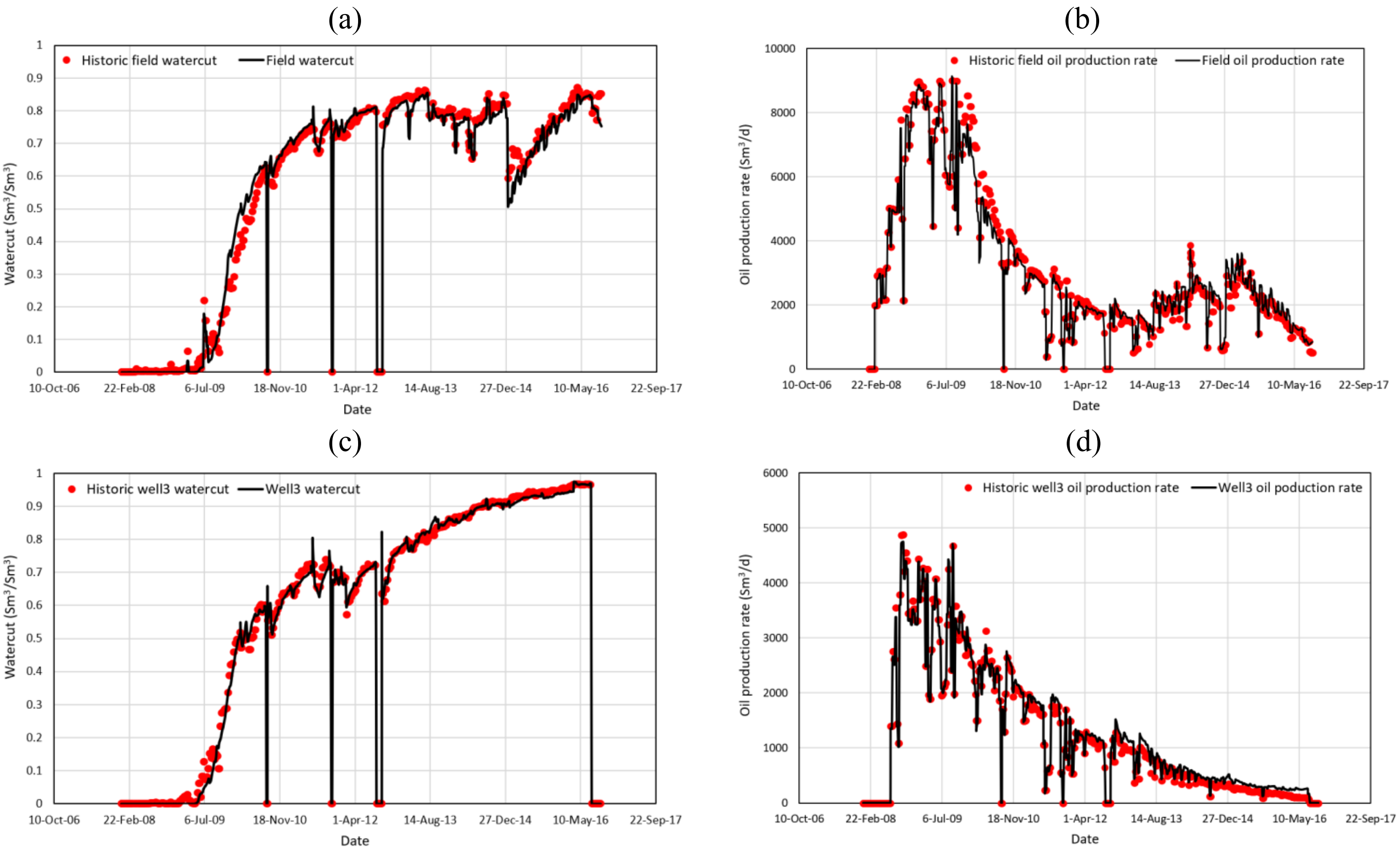


Figure 29. Volve reservoir model showing the water saturation

Table 15. Key characteristics of the Volve model

Parameter	Value	Unit
Size	2×3	Km^2
Number of cells	108×100×63= 680,400	-
Total proven oil	27.5	million Sm^3
Net-to-gross	93%	-
Average porosity	21%	-
Average permeability	1	Darcy
Average water saturation in the oil zone	20%	-
Gas-oil ratio	111-157	Sm^3/Sm^3
Formation volume factor	1.33-1.45	m^3/Sm^3
Maximum water injection rate	16,000	Sm^3/d
Maximum oil production rate	9,000	Sm^3/d
Maximum water production rate	10,400	Sm^3/d
Maximum liquid production rate	13,000	Sm^3/d

The reservoir includes five producers and three injectors, and it was produced between 2008 and 2016. The model is history matched in this eight-years interval. Figure 30 demonstrates the history matched watercut and oil production rate of the model in this period of time as examples.


(a)

(b)

(c)

(d)

Figure 30. History match for (a) field watercut, (b) oil production rate, (c) well 3 watercut, and (d) well 3 oil production rate

In this work, oil production is extended for ten more years (i.e., to Jan 2027) under a waterflooding scenario. Results for different parameters related to the grids and wells are extracted every six months to form the datasets for the grid-based and well-based SPM. The main goal of this work is to construct a well-based SPM and optimize production; however, we have implemented a hybrid SPM that requires the grid-based SPMs to work in parallel. The grid-based SPMs help to extract some dynamic parameters related to the grids and transfer them to the well-based SPM.

4.5. Results and Discussion

4.5.1. Sequential LHS Design

The reservoir includes five production wells, and LPR of each well is recognized as a designing parameter for production optimization. The LPR of the individual producers varies between zero and 5,000 Sm^3/d. As discussed in the SPM workflow section, the LHS design with GA was chosen to perform the sequential sampling. We began the initial design with ten sample points based on our experience. However, the underlying models (ANN and 1D-CNN) generated by only ten sample points failed to provide accurate results in the verification step with blind tests (further discussed in model training). As a result, we added five more points to the initial sample points based on the optimized algorithm to increase the accuracy. Table 16 presents a summary of these sample points.

Table 16. Initial and augmented sample points for sampling in Volve case study

	Sample point	LPR (Sm^3/d) well#1	LPR (Sm^3/d) well#2	LPR (Sm^3/d) well#3	LPR (Sm^3/d) well#4	LPR (Sm^3/d) well#5	LPR (Sm^3/d) Total
Initial sample points	1	68	1,521	2,632	3,665	4,027	11,912
	2	2,142	4,936	1,268	4,450	1,942	14,739
	3	1,956	2,416	4,764	3,134	323	12,594
	4	4,964	3,812	4,282	2,601	3,153	18,812
	5	3,336	3,098	882	1,688	4,887	13,892
	6	1,434	848	321	2,305	1,386	6,293
	7	4,118	1,084	2,001	4,915	2,003	14,121
	8	2,812	134	3,894	1,308	3,966	12,115
	9	866	4,257	3,473	921	2,850	12,366
	10	3,781	2,911	1,502	240	777	9,210
Additional sample points	11	364	423	4,484	596	412	6,279
	12	4,470	4,558	605	4,032	4,589	18,255
	13	1,218	3,407	3,205	3,697	1,114	12,642
	14	3,278	1,970	1,981	1,536	3,467	12,231
	15	2,653	2,091	2,815	2,972	2,458	12,989

Figure 31 shows the field cumulative oil production at Run 1,4, 6,7, and 10 between 2017 and 2027. This figure clearly justifies the selection of liquid production rates to have a good case for production optimization. For example, run 4 has the highest total liquid production rates (obtained from Table 16) among all sample points, but it did not have the highest cumulative oil production at any time in these ten years. Similarly, run 6 used the sample points with the lowest produced liquid; however, it produced more cumulative oil than run 1, which had twice the liquid production. Furthermore, by observing the curve for run 10, we cannot conclude that if the field is producing low in the initial years, it will continue with that trend to the final years. Run 10 initially had the lowest cumulative oil production, but it was the highest-producing scenario in the final year.

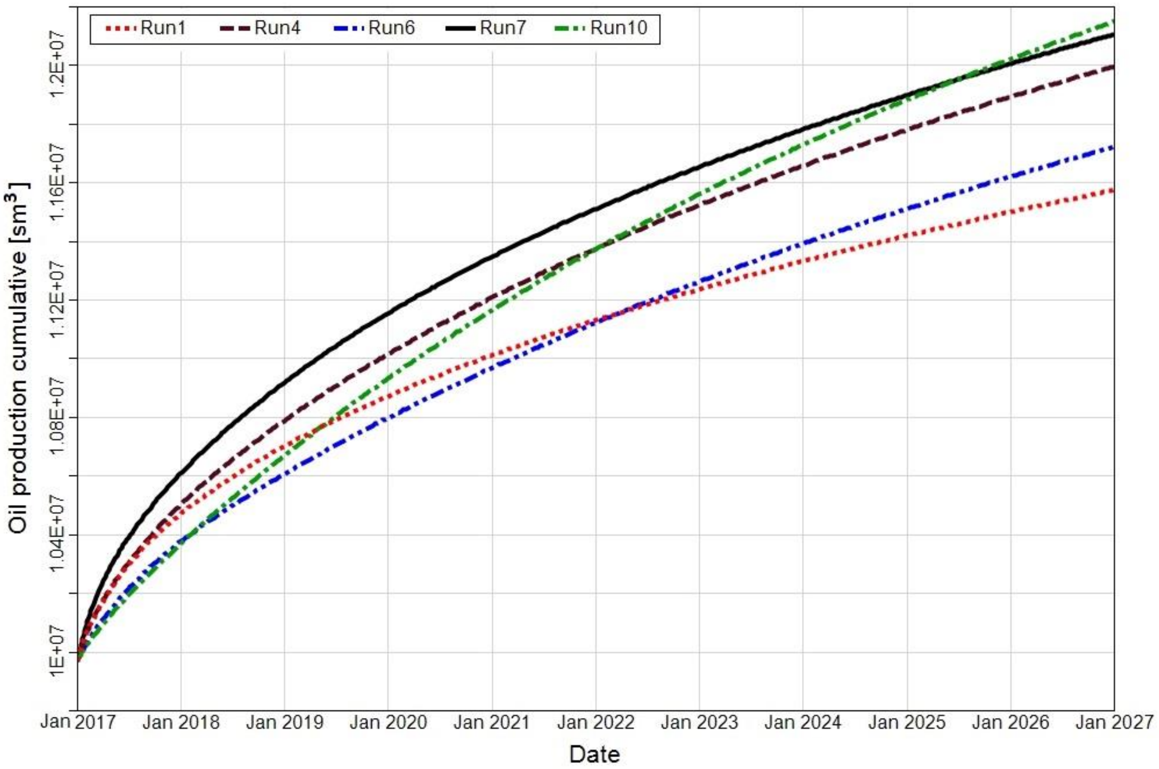


Figure 31. Field cumulative oil production at five sample points

4.5.2. Dataset Preparation

To construct the well-based SPM, various static and dynamic parameters corresponding to the wells were extracted. This dataset consists of the initial design parameters and the obtained parameters from the numerical model's run. Table 17 lists the parameters that were used for the well-based dataset. It should be noted that the LHS design parameters refer to the LPRs, and they are the rates at which the producers start to produce liquid.

A tiering system was introduced to increase the model's accuracy and find new patterns in the well-based dataset. Each tier contains a layer of grids around the wells, as demonstrated in Figures 32 and 33. For example, tier one only consists of the grids that the wellbore passes through, and tier two consists of the grids in the wellbore plus the next layer of grids, and so on. Consequently, various static and dynamic parameters were averaged for each tier and added to the dataset.

Table 17. Static and dynamic parameters extracted for the well-based dataset

Parameter	Details
Static	
LHS design parameters	LPRs for producers
Distances to 1st, 2nd, and 3rd producers	At four different vertical depths
Distances to 1st, 2nd, and 3rd injectors	At four different vertical depths
Well index	
Well type	Injector, producer
Porosity	Tier 1 to tier 5
Permeability	Tier 1 to tier 5
Transmissibility	Tier 1 to tier 5
Dynamic	
Oil saturation	Tier 1 to tier 5
Pressure	Tier 1 to tier 5
Cumulative oil production	Individual wells

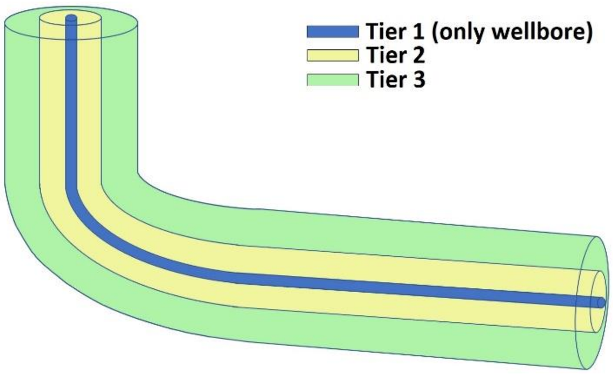


Figure 32. Tiering system for the wells

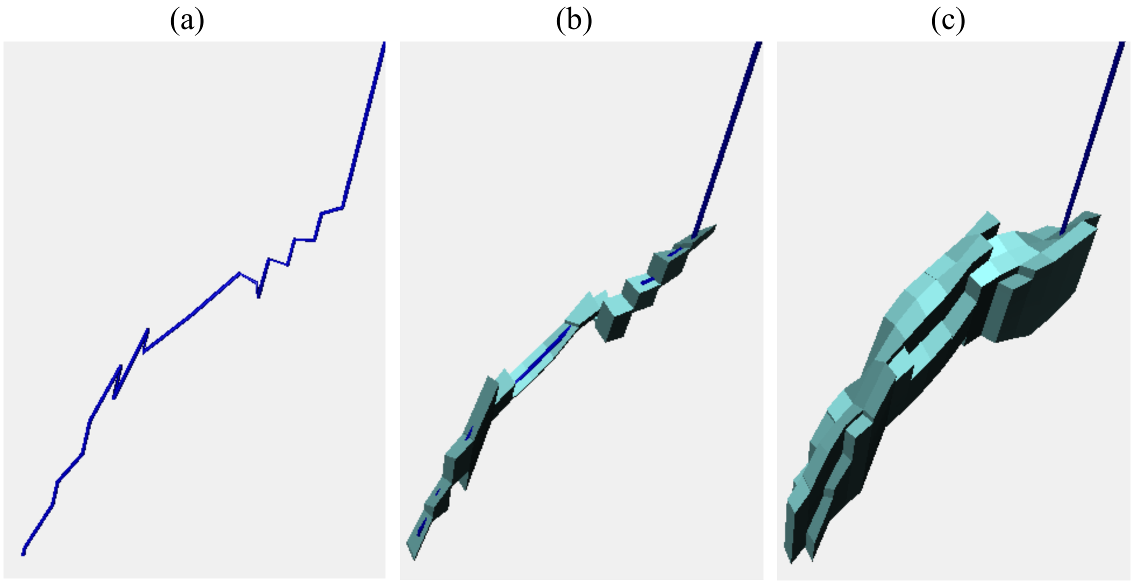


Figure 33. Tiering system for (a) Well#3, (b) tier 1, and (c) tier 2

The averaged dynamic parameters (oil saturation and pressure) for the grids in each tier were obtained using a hybrid implementation of grid-based SPM with well-based SPM. Further details on the construction of the grid-based SPM can be found in the paper presented by Bahrami and James [231]. Figure 34 demonstrates the workflow for the hybrid implementation of well-based and grid-based SPMs.

The configuration of the well-based dataset is shown in Figure 35. The number of rows equals 20×15n, where n is the number of producers. Also, the number of columns is 62, of which 61 are the inputs, and one corresponds to the outputs listed in Table 17.

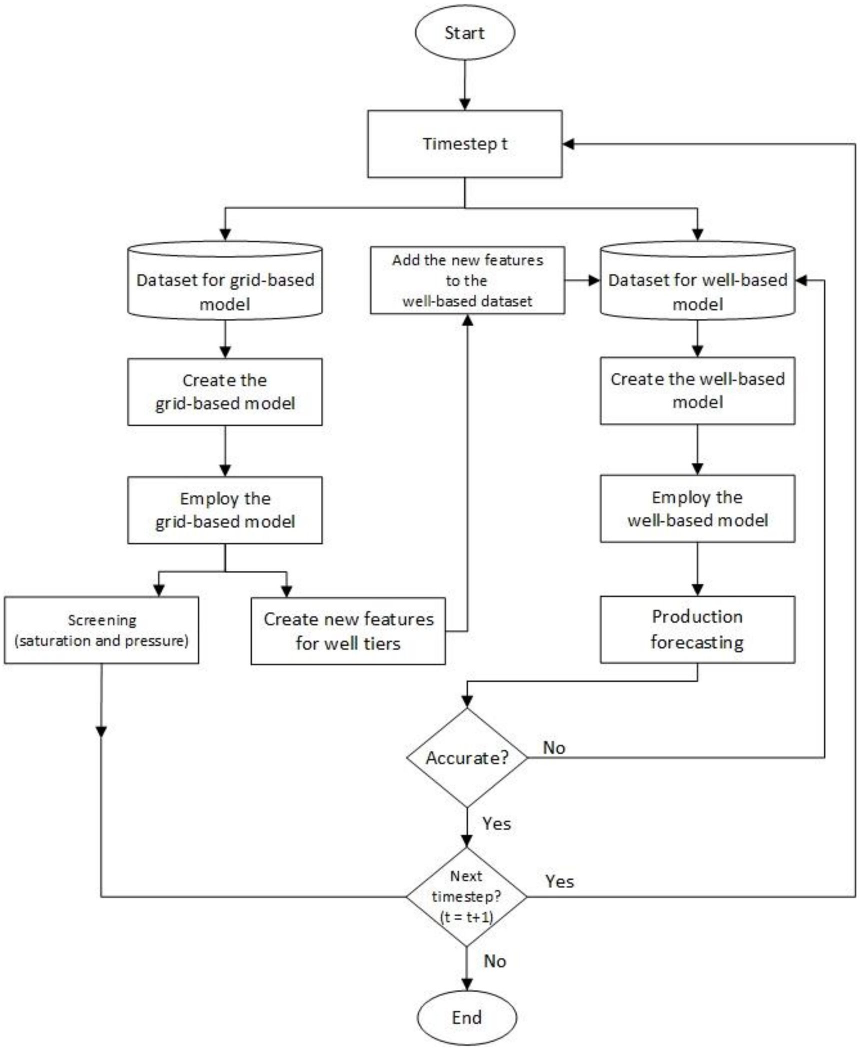

Start
Timestep t
Dataset for grid-based model
Add the new features to the well-based dataset
Dataset for well-based model
Create the grid-based model
Create the well-based model
Employ the grid-based model
Employ the well-based model
Screening (saturation and pressure)
Create new features for well tiers
Production forecasting
Accurate?
No
Yes
Next timestep? (t = t+1)
Yes
No
End

Figure 34. Workflow for parameter exchange between well-based and grid-based SPMs

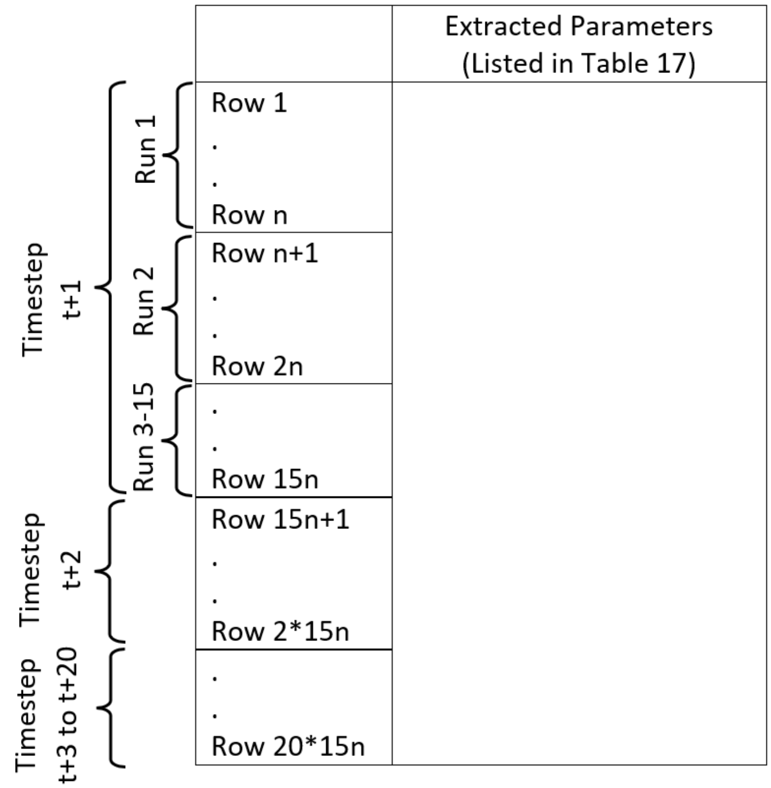


Figure 35. Well-based dataset configuration

4.5.3. Feature Ranking and Feature Selection

To find a generalized ranking of the all the parameters, the *fscaret* package was used. We used 23 different predictive models, listed in Table 3, to determine the importance of each input parameter. The individual predictive model's importance was then averaged and scaled to 100 to rank the parameters. Figure 36 demonstrates the ten parameters with the highest average importance when the output is the cumulative oil production.

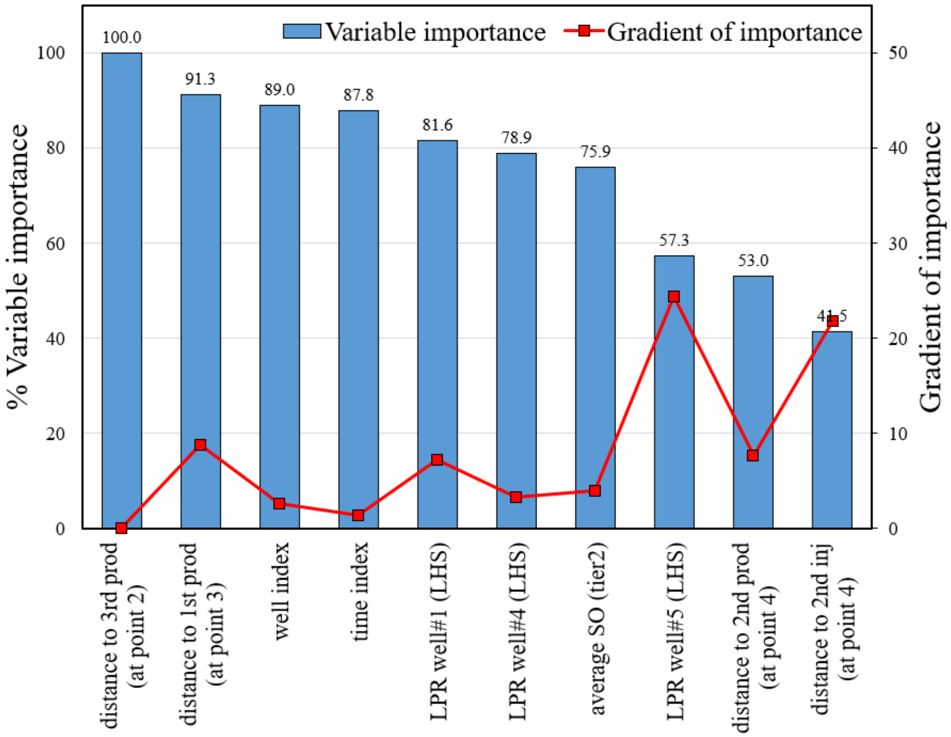


Figure 36. Parameters' importance and gradient of importance on cumulative oil production

The variable shown in this figure may not necessarily have a relationship with the physics of the problem. For example, it is assumed that the designing parameters (LPRs) are the most influential parameters, but in the formed dataset, LPR well#1 is placed in the fifth rank, and other LPRs are in lower ranks. If LHS generates other sample points, the parameters' importances may differ from the current ranking order. Nonetheless, this figure presents the important parameters for building an underlying model in the next step. The *fscaret* package is a filter feature selection method, and it helps find the best subsets of parameters to train the underlying model. These subsets can be determined by having the gradient of importance shown in Figure 36. Based on this figure, wherever we have a big gradient of importance, the parameters prior to that point can be selected as a subset. Accordingly, four reduced input vectors containing 7, 9, 15, and 19 parameters were selected to train the underlying models with

ANN and 1D-CNN. Finally, the models' goodness of fit and accuracy on each subset were checked to select the optimal subset.

4.5.4. Model Training and Validation

We trained and tested the ANN models at different subsets of 7, 9, 15, and 19 parameters obtained in the feature selection step. Table 18 displays the goodness of fit and accuracy for these subsets. The vector of input parameters holding 15 parameters was selected as the best subset for further improvements. This subset was selected because it was determined that keeping more parameters in the dataset did not add much value to the model, and the same goodness of fit and accuracy could be achieved with a lower number of parameters. The subset with 15 parameters contained all the design parameters, indexing parameters, and eight other generated parameters.

Table 18. ANN models' goodness of fit and accuracy at different subsets

Test	Selected subset	Test set		
		R^2 (%)	RMSE (Sm^3)	MAPE (%)
1	LHS design parameters + index parameters	92.65	456E+3	3.18
2	First 7 parameters from feature selection	97.16	228E+3	1.94
3	First 9 parameters from feature selection	99.21	94E+3	0.68
4	First 15 parameters from feature selection	99.63	54E+3	0.31
5	First 19 parameters from feature selection	99.68	50E+3	0.28
6	All parameters	99.59	60E+3	0.32

The 1D-CNN model was also trained on the same dataset, including 15 parameters, to compare it to the ANN model. The best results obtained in a 1D-CNN model consist of two convolutional layers (each containing 50 filters and kernel size equal to four) and two fully connected layers (each containing 60 neurons). The goodness of fit for the 1D-CNN model was similar to the ANN model (R^2 = 99.61, and RMSE=36E+3), but the 1D-CNN accuracy (MAPE = 0.23%) was slightly improved compared to the ANN model.

We also performed an analysis to check how a traditional proxy model would behave if constructing an SPM was not an option. Accordingly, an ANN was trained on a dataset consisting of only the design input parameters (i.e., LPRs for individual wells), the indexing parameters, and the output (test 1 in Table 18). We conducted a grid search to find out the best ANN hyperparameters using 10-fold cross-validation. The ANN with two hidden layers, each containing 100 neurons, Rectified Linear Unit (ReLU) activation function, and batch size equal to two, produced the best goodness of fit. The R^2 and RMSE for the test set were 0.9265 and 456E+3, respectively. Consequently, it can be concluded that constructing the SPM could raise the accuracy by 3% compared to the TPM trained in this case study. The difference between SPM and TPM performance can be very significant in grid-based cases. In another study carried out over grid-based SPM on the Volve model [231], the goodness of fit was approximately 94-98%. However, the TPM was overfitted on the same dataset.

In a final effort to validate the model on unseen data, random LPRs were introduced (Table 19). Then, to generate the input vector to be fed into the 1D-CNN model, the grid-based SPM was run in parallel to extract the parameters relating to the well tiers. Finally, the 1D-CNN models' goodness of fit and accuracy were obtained. The results show that for three blind tests, R^2 was above 99%, and MAPE was below 1%. Figure 37 demonstrates the performance of the model on blind test 3 as an example.

Table 19. Blind tests and the goodness of fit for the 1D-CNN model

	LPR (Sm^3/d) well #1	**LPR (Sm^3/d) well #2**	**LPR (Sm^3/d) well #3**	**LPR (Sm^3/d) well #4**	**LPR (Sm^3/d) well #5**	**R^2 (%)**	**RMSE (Sm^3)**	**MAPE (%)**
Blind Test 1	4,399	2,605	308	4,149	4,782	99.38	72E+3	0.52
Blind Test 2	1,873	3,389	3,614	4,532	1,528	99.28	81E+3	0.60
Blind Test 3	1,937	273	3,720	2,257	4,777	99.31	80E+3	0.59

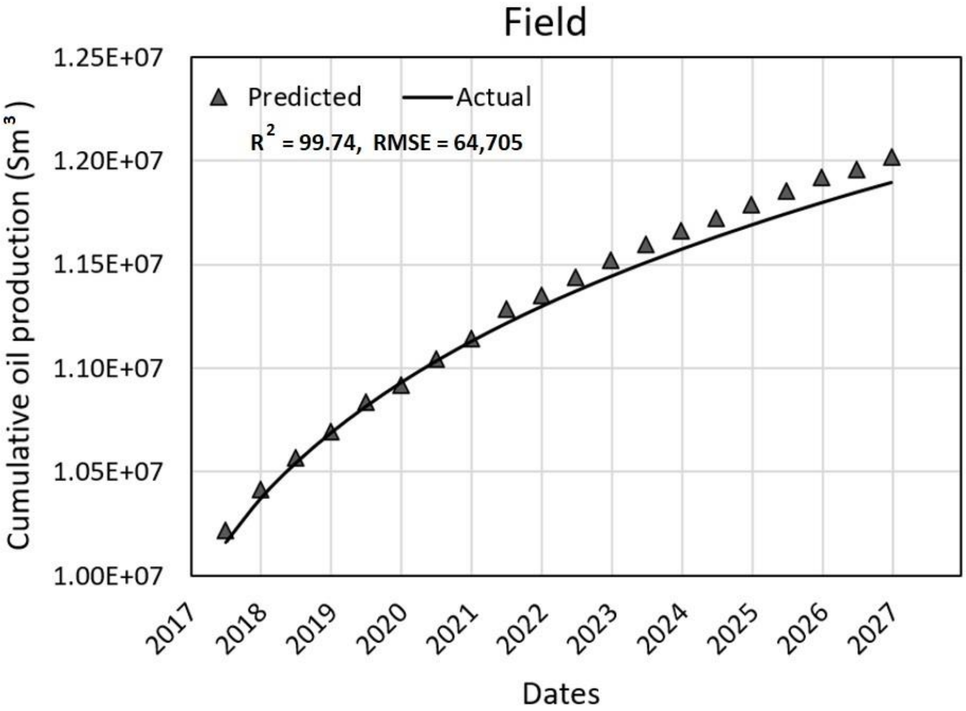


Figure 37. Cumulative oil production for the Volve model at blind test 3

It is worth mentioning that we did all the procedures to construct the SPM with ten sample points before applying 15 sample points. However, at the validation step using the blind test, the R^2, RMSE, and MAPE were obtained at 93.20, 411E+3, and 3.04, respectively. As a result, five more sample points were added to the initial design to increase the prediction ability of the SPM.

4.5.5. Optimization

The presence of an objective function is necessary for every optimization problem. This work uses the SPM as our fitness function to evaluate the individual well's cumulative oil production at the end of the production period. Moreover, the objective function is the summation of the cumulative oil production of all wells.

The constrained maximization optimization problem of this work is as follows:

$$\begin{aligned} &maximize \quad \sum_{i=1}^{n} COP_i \\ &subject\ to \quad \sum_{i=1}^{n} LPR_i < 13{,}000 \end{aligned} \tag{17}$$

where n is the number of wells, COP is the cumulative oil production, LPR is the liquid production rate (optimized parameters), and 13,000 is the maximum LPR allowed to be produced for the field. In every iteration of PSO and GA, the SPM works as the fitness function and predicts the cumulative oil production. One of the drawbacks of using the LHS design of experiment is that it misses having sample points at the parameters' minimum and maximum values. This makes the SPM extrapolate at the minimum and maximum points. The range for the designed parameters (LPRs) was set between 0 and 5,000 Sm^3/d. To avoid the SPM from extrapolation, the range for optimized parameters in the optimization process was 100 Sm^3/d shortened. Consequently, the lower and upper bounds for the optimized parameters were selected as 100 and 4,900 Sm^3/d, respectively.

There are some parameters that need to be set in PSO. One is the number of particles in the swarm. The initialization of the swarm is very important to cover the search space and find the optimum more efficiently. If a low-intensity swarm is selected, the algorithm may be trapped in the local optimum. We selected 200 particles to be distributed randomly in the parameters' domain space. However, this problem is a constrained optimization, and it is important always to meet the constrained condition. This means the condition should be met for the initial distribution of the particles and updates to the particles' velocity and position. The initial distribution can be controlled simply by dropping the particles that violate the condition. Hence, we started with 1,000 random particles. Then, 200 particles were kept from the ones that met the condition. A penalty term was also added to the objective function to control the updated particles. The penalty term prevents the particles from violating the introduced condition of the problem. This penalty term was selected as a big negative value to help the algorithm avoid the violated domain in the subsequent iterations. Another parameter in the PSO algorithm is the stopping criteria. We used 40 iterations as the stopping criteria. At each iteration, each particle that is a vector of the optimizing parameters was fed into the SPM, and the cumulative oil production was evaluated. So, the total number of fitness function evaluations was 200 for one iteration and 8,000 for all iterations. In this study, values of inertia, and acceleration coefficients c_1 and c_2 were kept constant at 0.7, 0.7, and 1.4, respectively.

PSO is a stochastic algorithm, and one realization or run may not provide a confident final result. In this work, five realizations based on random particles' positions were introduced. Table 20 presents the best particle values and the global optimum at different PSO realizations. Figure 38 displays the cumulative oil production versus iterations at PSO realizations in the last timestep.

Table 20. Summary of random PSO realizations in waterflooding optimization

Realization	LPR (Sm^3/d) well #1	LPR (Sm^3/d) well #2	LPR (Sm^3/d) well #3	LPR (Sm^3/d) well #4	LPR (Sm^3/d) well #5	LPR (Sm^3/d) Total	Global best value (cumulative oil production)
1	2,542	100	3,986	100	4,900	11,628	1.232E+7
2-5	2,491	1,223	3,873	100	4,900	12,587	1.238E+7

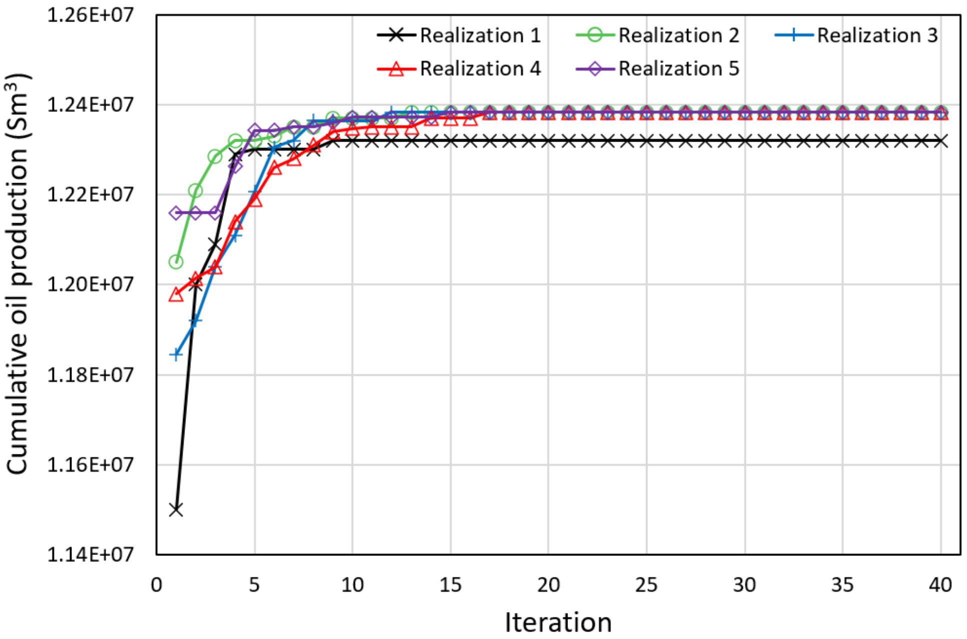


Figure 38. Cumulative oil production vs. iteration at different PSO realizations in waterflooding optimization

For the GA optimization, the same objective function and constraint were used. The optimization started with a population size of 200 solutions, and the stopping criteria was set at 40 iterations. The mutation rate and crossover rate of 0.02 and 0.8 were used to generate the new offsprings. The total fitness function evaluations in GA optimization were, similar to the PSO. We repeated the GA process five times to check how the stochastic nature of this algorithm would change the results. Table 21 shows the best particle values, and the global optimum at different GA realizations. Figure 39 illustrates the cumulative oil production versus iterations at GA realizations in the last timestep.

Table 21. Summary of random GA realizations in waterflooding optimization

Realization	LPR (Sm^3/d) well #1	LPR (Sm^3/d) well #2	LPR (Sm^3/d) well #3	LPR (Sm^3/d) well #4	LPR (Sm^3/d) well #5	LPR (Sm^3/d) Total	Global best value (cumulative oil production) (Sm^3)
1,4	2,491	1,223	3,873	100	4,900	12,587	1.238E+7
2	1,690	1374	4,002	100	4,900	12,066	1.234E+7
3, 5	2,542	100	3,986	100	4,900	11,628	1.235E+7

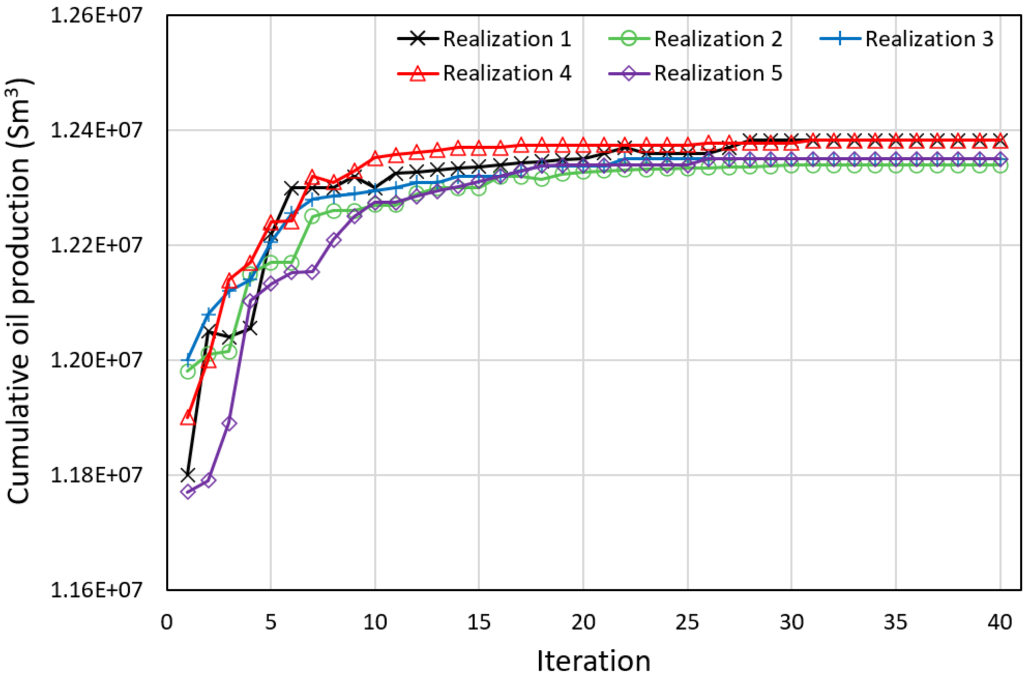


Figure 39. Cumulative oil production vs. iteration at different GA realizations in waterflooding optimization

Based on the results, both PSO and GA found the global optimum at input vector of (2491, 1223, 3873, 100, 4900). GA was able to find the global optimum two times out of five realizations. In GA, the convergence to the final result happened in fewer than 30 iterations. On the other hand, PSO converged the results to optimum in fewer than 20 iterations. Furthermore, in five realizations of PSO, only one failed and was trapped in the local optimum. Consequently, it can be concluded that with five optimizing parameters, PSO was more confident in finding the global optimum, and it converged to a solution in fewer iterations than GA. Also, the calculation time for the PSO was slightly faster than for GA.

The time spent to do the optimization with PSO and GA was approximately 11 to 11.5 hours for one realization on a machine with 16 GB RAM, and 3.20 GHz processor. The SPM was applied 8,000 times in this time frame, meaning each SPM deployment took only five seconds. One numerical run on the same machine took approximately five hours. So, it would take years to finish the task if one wanted to apply a similar optimization algorithm on the numerical model. It should be noted that the Volve model is a small case with limited wells. The difference can be enormous in a bigger reservoir or numerical compositional model. Implementing SPM can save a huge amount of time, justifying the effort to construct the SPM.

It should be noted that the SPM used in the optimization process can handle cases in that producers have varying production rates in the determined period of time. We started the simulation runs based on the given LPRs (from LHS design of experiment); however, it does not mean that the producers continue to produce fluid at the introduced LPR values. The individual wells and field may decline in liquid production for many reasons, such as constraints applied to the injectors or producers. The newly generated dataset in this work includes the production data for cases where producers continue to produce liquid at fixed or varying values. Consequently, the constructed SPM can capture both cases, and optimizers are viable even when the production rates decline.

4.6. Conclusions

In this work, we developed an SPM for use as an approximation of the Volve reservoir model. This method was first introduced by Mohaghegh [7]. SPM can quickly reproduce the results from the reservoir numerical model without losing much underlying reservoir information. SPM works based on machine learning and pattern recognition techniques that can uncover unseen patterns within the numerical model.

The objective of this paper was to develop a well-based SPM to mimic the well/field cumulative oil production and to optimize the field production. The implemented SPM was further modified in sampling, feature engineering, and feature ranking steps. Furthermore, the applicability of the 1D-CNN model as the underlying model was tested in the SPM construction workflow.

In the sampling step, we used the sequential sampling approach. LHS sampling was the main algorithm in this step that was coupled with GA to find the optimum sample points. Later, we added five more sample points to the initial points to prohibit resampling from the beginning. As one of the main goals of constructing the SPM is to speed up the process of getting results from the numerical model, sequential sampling helped to save a lot of time. After running the numerical model at sample points, we feature-engineered and extracted various static and dynamic parameters directly from the wells or the introduced tiering system. The tiering system was the wells' surrounding grids that helped obtain new averaged parameters such as oil saturation, pressure, porosity, etc.

Furthermore, we used an average feature ranking technique using the *fscaret* package to find the best subsets of parameters for the feature selection step to reduce the well-based dataset's size. Only 15 out of the total 61 input parameters were chosen based on the feature selection process. The average feature ranking provided a more confident ranking for the parameters. In the underlying model step, we trained both ANN and 1D-CNN techniques on the selected 15 parameters. The goodness of fit for the 1D-CNN model was similar to the ANN model, but its accuracy (presented in MAPE) was slightly better than ANN.

Finally, we implemented the PSO and GA optimization algorithms to find the best selection of designing parameters (individual well’s LPR) and to maximize the cumulative oil production over ten years. Both optimizers were quite successful in finding the global optimum. Nevertheless, PSO showed a more reliable and faster convergence to the solution.

Chapter 5. Screening and Production Optimization in A Water-Alternating-Gas (WAG) Process

5.1. Introduction

Water-alternating-gas (WAG) is a tertiary oil recovery technique that requires injecting water and gas into the reservoir. WAG is generally performed when the primary and secondary recovery techniques do not enhance the oil production. The main objective of WAG is to increase the reservoir pressure and sweep more oil toward the producers. This happens by injecting a slug of gas for a period of time, followed by a slug of wag. In WAG, the sweep efficiency usually increases by two mechanisms of microscopic and macroscopic displacement [232]. Microscopic displacement efficiency refers to dissolving the gas slug into the oil at the microscopic scale, which helps mobilize the remaining oil. On the other hand, macroscopic displacement efficiency that is controlled by water injection pushes the miscible slug forward to the producers. Two common classes of WAG are miscible and immiscible injection processes. In miscible WAG, the reservoir pressure is kept above the minimum miscibility pressure of the fluids, and all proportions of gas can be mixed in the oil and swell the remained oil. Miscible WAG can usually result in a higher recovery than an immiscible WAG. The main objective of an immiscible WAG is to stabilize the front and improve contact with the unswept areas. Many factors can affect the WAG process, such as reservoir heterogeneity and characteristics, rock and fluids characteristics, injected gas composition, injection pattern, WAG ratio, and injection rates [233], [234].

The objective of this chapter is to develop grid-based and well-based SPMs for the immiscible WAG process with the same methodology introduced in chapters 3 and 4. The grid-based and well-based SPMs can then be applied to screen and optimize the oil production for the WAG process, respectively. Among the effective parameters in WAG optimization, the parameters related to reservoir heterogeneity, rock and fluids characteristics, injected gas composition, and well patterns are fixed. Nevertheless, other operational parameters such as gas/water cycles, gas/water injection rates, distribution of injecting fluids between the injectors, and BHP of producers are considered as influencing parameters for WAG design. The WAG cycle can vary in value from days to months, and it depends on reservoir heterogeneity, well patterns, and injection and production volumes [235]. The volumes of the injected fluid and recovered oil also depend on the fluids' injection rate at their cycle length and the BHP of the producers. Designing a WAG process and optimizing oil production can sometimes be very challenging when incorporating all the controlling parameters. This chapter aims to construct and implement the grid-based and well-based SPMs to ease the process of WAG screening and optimization.

5.2. Case Study

Norne is a sandstone reservoir located in the Norwegian sea discovered in December 1991. It was in production between November 1997 and December 2006. The field had an original oil in place of 157 million Sm^3. The Norne structure consists of four main sections C, D, E, and G, in which nearly 98% of the original oil is placed in segments C, D, and E. Figure 40 shows the Norne field's different segments.

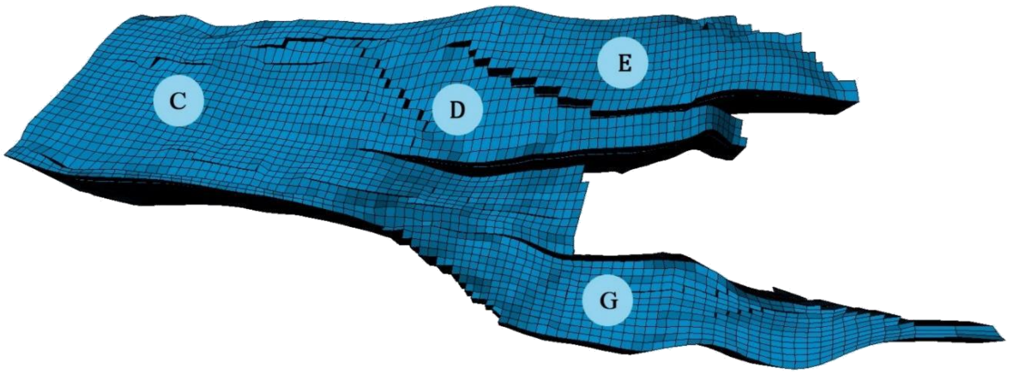


Figure 40. Different segments of the Norne field

The Norne reservoir model contains 46×112×22 grids in which 44,927 are active. In this work, the history-matched E-segment reservoir model is used as the case study to construct the SPM. The average porosity is 0.24, the permeability is 330 mD, and the NTG is 0.87 for the E segment. Wells F-1H and F-3H are the injectors, and wells E-2H, E-3H, and E-3AH are the producers for the E-segment. The well location and paths in the Norne E-segment are shown in Figure 41.

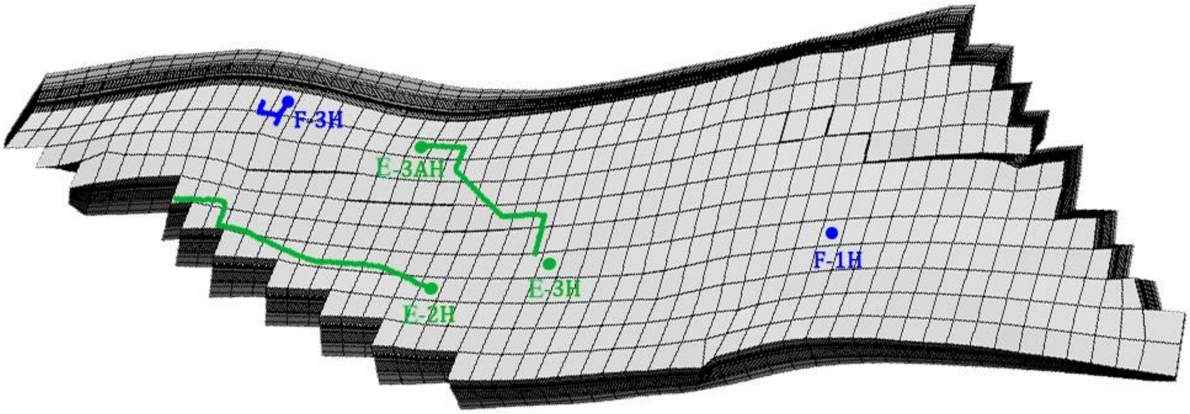


Figure 41. Well locations and their paths in Norne E-segment reservoir model

5.3. Results and Discussion

5.3.1. Sequential LHS Design

Sampling started with choosing nine parameters of the gas/water injection cycle, field gas/water injection rate, gas/water injection distribution between two injectors, and BHP for production wells E-2H, E3H, and E-3AH. Table 22 presents the WAG parameters and the selected ranges for the design of experiment. The minimum and maximum values for the injection rates and the BHPs were chosen based on the production history. The WAG injection, in this case, was started by injecting a gas cycle since the injectors were under water injection in 2005, which was the starting time for the WAG process. The injection cycles were chosen between 3 and 24 months. The fluid distribution ratio between wells varies between zero and one, in which fluid distribution equal to zero means that all fluid is injected by well F-3H, and one refers to injection only by well F-1H.

Table 22. WAG parameters with their ranges

Parameter	Range	Unit
Gas cycle	3-24	months
Water cycle	3-24	months
Field gas injection rate	3-9	MMSCM/d
Field water injection rate	10-32	MSCM/d
Gas distribution ratio	0-1	
Water distribution ratio	0-1	
BHP (Well E-2H)	100-260	bar
BHP (Well E-3H)	100-260	bar
BHP (Well E-3AH)	100-240	bar

The design started with 20 sample points based on the rule of thumb and the experience of the previous Volve case study. In Volve case, we had five parameters, and the initial sample points number was equal to 10. Similar to previous chapters, the LHS algorithm coupled with GA was used to perform the initial design. Also, five more sample points later were added based on the optimized augmentation algorithm to fill the empty domains of the parameters and increase the underlying model accuracy. Table 23 presents a summary of these sample points and a blind sample point. This blind sample point was used for the underlying model validation. The WAG process continued for seven years, between 2005 and 2012. Figure 42 shows the gas/water injection cycles for all the sample points during this period. We considered the field gas/water injection rates as the design parameters in the sampling. The individual well's gas/water injection rate was calculated using the gas/water distribution ratio between wells. Another important parameter in the WAG process is the WAG ratio, which is the ratio of the injected water volume to the injected gas volume in their cycles. The WAG ratio was not a design parameter in this work and was calculated based on other parameters. Figures 43 and 44 illustrate the relationship between water and gas injection rates for well F-1H and F-3H. The diameter of the points in these figures demonstrates the WAG ratio magnitude. The higher values of the injected water volume and the lower values of the injected gas volume result in higher WAG ratios. As a result, the WAG ratios are smaller on the lower right side of the figures, and the points at the top left side are bigger.

Table 23. Initial and augmented sample points for sampling in Norne case study

	Sample point	Gas cycle (month)	Water cycle (month)	Gas inj. rate (well F-1H) (MMSCM/d)	Gas inj. rate (well F-3H) (MMSCM/d)	Water inj. rate (well F-1H) (MSCM/d)	Water inj. rate (well F-3H) (MSCM/d)	BHP (well E-2H) (bar)	BHP (well E-3H) (bar)	BHP (well E-3AH) (bar)
Initial sample points	1	17.9	21.5	1.95	5.96	2.94	16.67	233	226	209
	2	9.3	7.8	1.88	1.13	20.48	0.68	154	134	130
	3	22.6	12.8	1.49	4.45	4.27	7.67	117	189	109
	4	6.3	23.9	1.12	5.46	14.23	2.01	162	142	198
	5	3.6	6.0	2.96	3.90	21.06	9.39	213	233	184
	6	21.5	10.2	0.38	4.91	15.25	8.85	240	111	167
	7	17.7	22.9	4.47	3.11	22.91	4.98	190	176	104
	8	10.4	16.2	7.79	0.80	6.77	7.58	139	239	194
	9	19.3	11.3	5.23	1.81	10.69	10.29	132	101	239
	10	8.7	19.1	4.82	0.74	15.29	12.07	249	148	231
	11	23.8	4.6	5.13	1.11	7.84	18.46	223	206	144
	12	20.3	14.0	1.86	1.89	16.97	1.20	184	252	225
	13	11.7	8.9	4.87	2.36	8.15	2.26	259	161	140
	14	5.2	11.9	4.42	3.73	0.52	22.37	168	122	124
	15	12.7	17.4	0.08	4.07	7.20	24.60	143	169	185
	16	14.1	15.6	4.54	0.17	10.12	19.32	104	217	162
	17	15.3	6.7	1.01	7.76	18.64	6.74	116	204	176
	18	16.1	20.2	2.85	0.71	0.94	14.41	199	131	156
	19	7.3	3.3	1.71	3.20	1.45	11.57	179	184	217
	20	4.7	17.9	1.65	2.56	7.29	10.23	204	253	114
Additional sample points	21	3.9	21.4	2.39	6.00	10.08	1.15	109	117	222
	22	21.2	5.2	3.17	0.26	12.69	9.25	246	258	134
	23	17.1	8.7	0.68	3.80	7.45	21.88	121	246	159
	24	11.2	16.9	1.16	3.83	9.73	5.23	135	153	148
	25	13.4	13.7	4.90	1.92	9.02	14.60	148	178	168

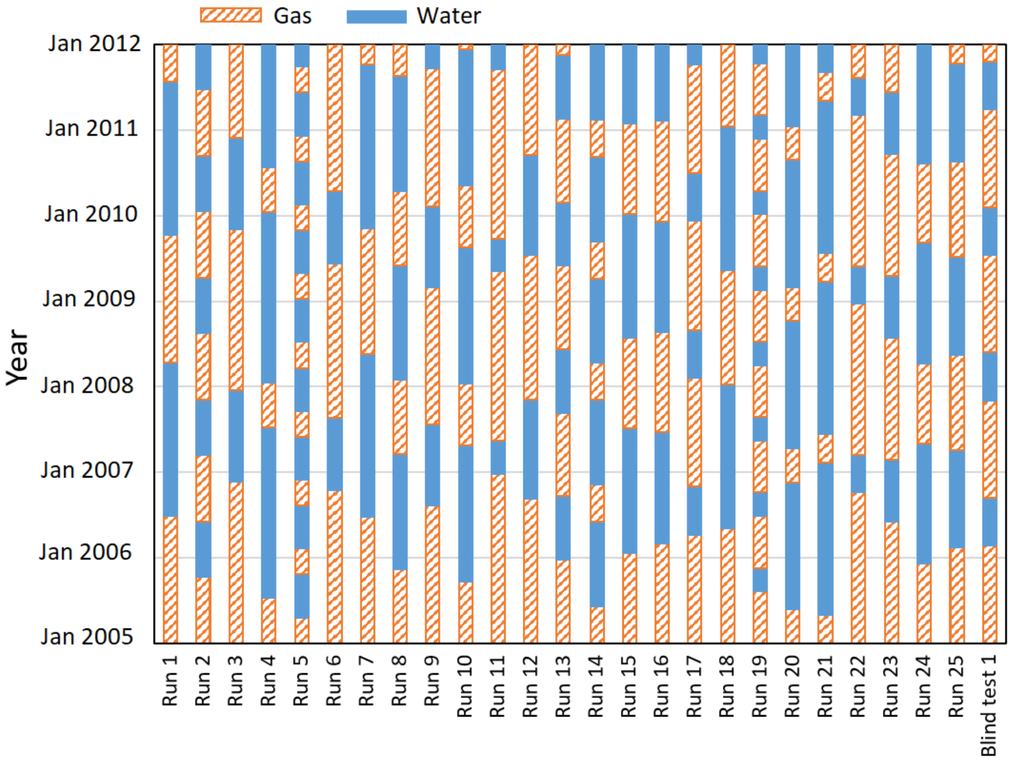


Figure 42. Gas and water injection cycles at all the sample points

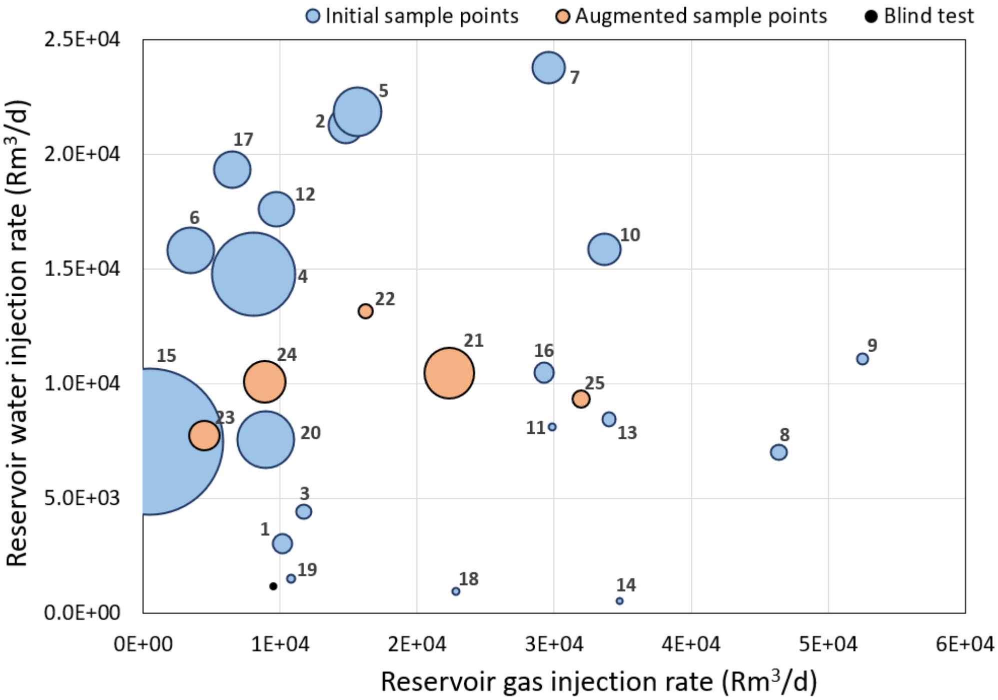


Figure 43. Gas injection rate vs. water injection rate for well F-1H at different sample runs

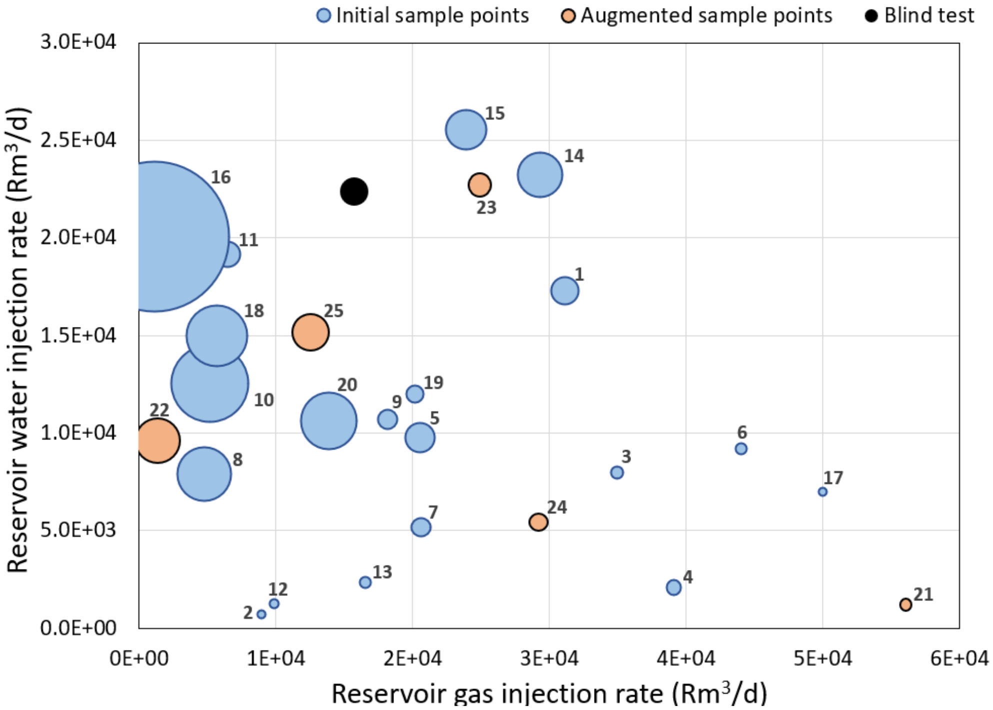


Figure 44. Gas injection rate vs. water injection rate for well F-3H at different sample runs

Figures 45 and 46 demonstrate the relationship between water and gas slug volumes at the sample points.

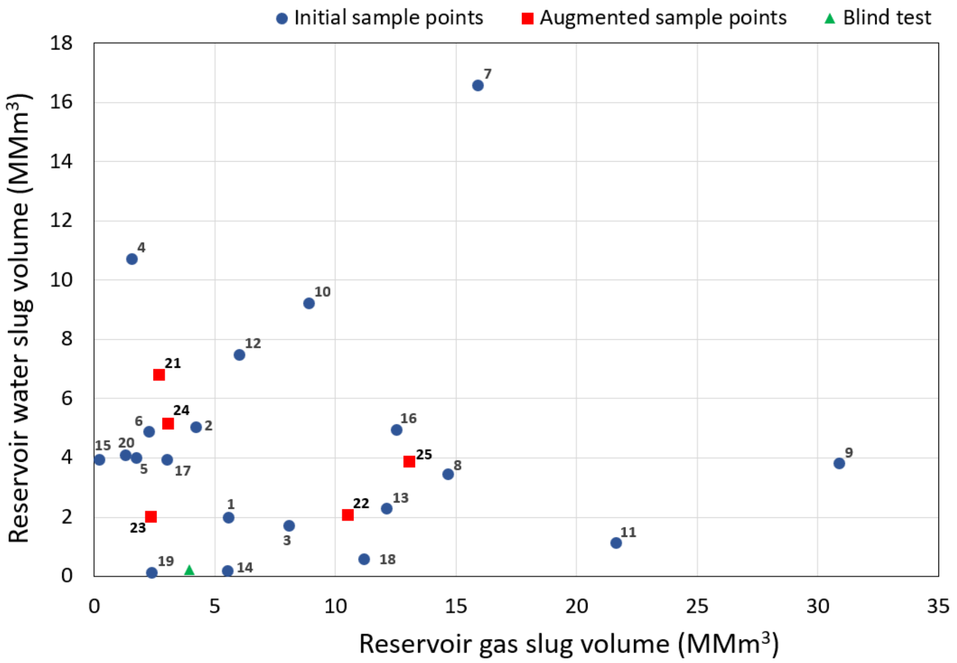


Figure 45. Gas slug volume vs. water slug volume for well F-1H at different sample runs

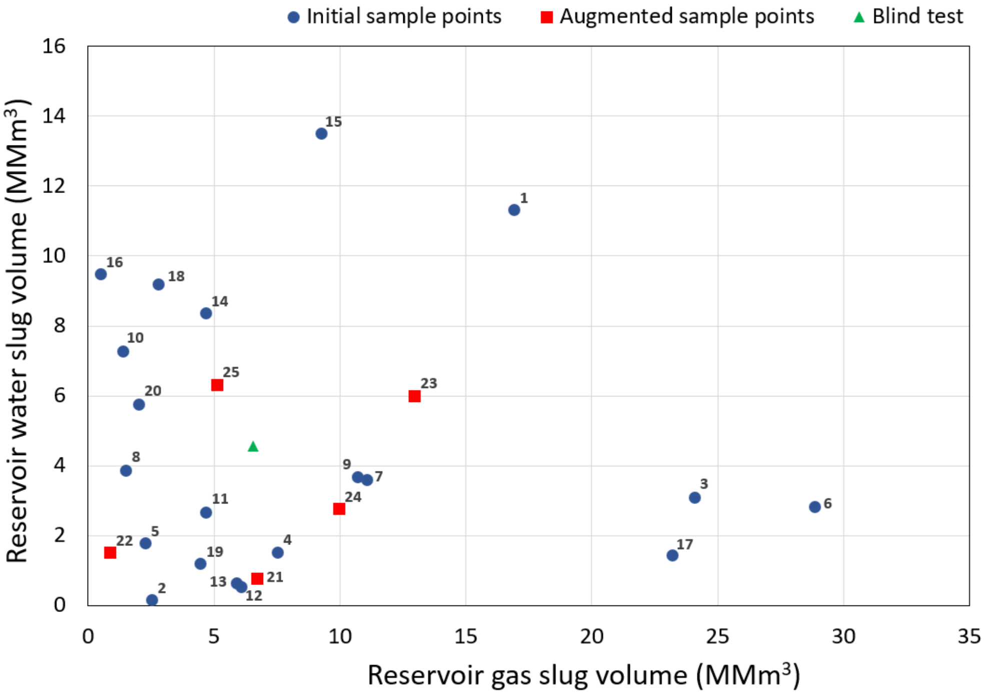


Figure 46. Gas slug volume vs. water slug volume for well F-3H at different sample runs

Figure 47 shows the injection and oil production rates at sample point 22, and Figure 48 illustrates the field cumulative oil production at some of the sample points as the examples.

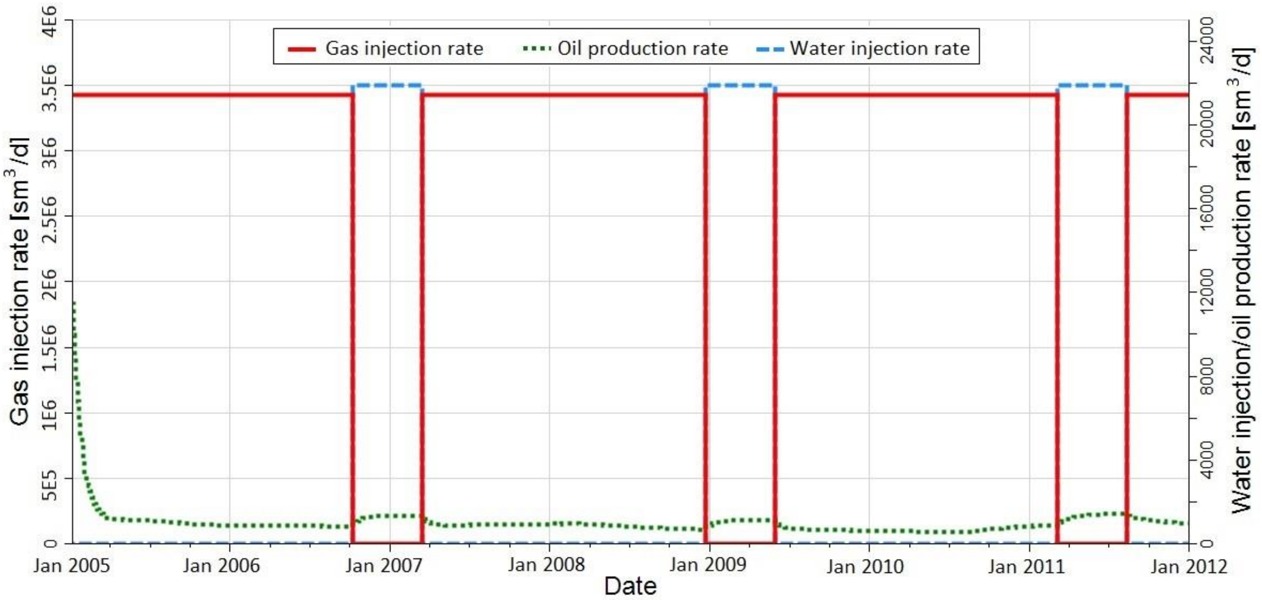


Figure 47. Injection and oil production rates at sample point 22

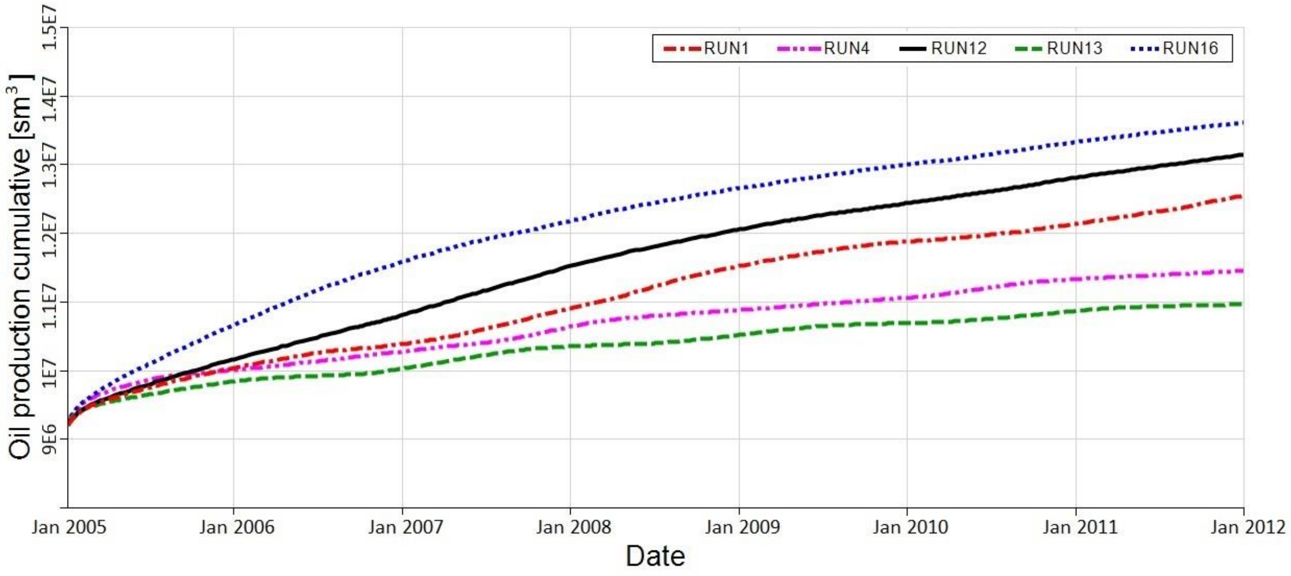


Figure 48. Field cumulative oil production between 2005 and 2012 at selected sample points

5.3.2. Dataset preparation

After running the simulator at the sample points, different static and dynamic data were extracted for the grid-based and well-based datasets, similar to the list introduced in Tables 6 and 17, with this difference that for the grid-based dataset, the outputs were the grids' oil, water, and gas saturations. The Norne field consists of 113,344 grid blocks, of which 8,922 grid blocks correspond to the E-segment. Consequently, the grid-based dataset contained 8,922×25 rows; each corresponds to one grid block. The number of columns in this dataset was equal to 62, of which 41 were the input parameters and 21 were the outputs at various time steps.

On the other hand, the well-based dataset had 21×25×3=1,575 rows (21 timesteps, 25 runs, and 3 producers) and 58 columns. As in Chapter 4, a tiering system was introduced, and the average of the grids' dynamic parameters within each tier was calculated. These new feature-engineered parameters related to the tiers eventually helped to increase the accuracy of the well-based SPM.

5.3.3. Grid-Based Underlying Model

The procedure to train the grid-based SPMs when the output is the grids's oil saturation is explained in this section. Similar procedures were also carried out to predict the grids' water and gas saturations. To find a generalized ranking of the all the parameters, the *fscaret* package was used. We used 23 different predictive models, listed in Table 3, to determine the importance of each input parameter. The individual predictive model's importance was then averaged and scaled to 100 to rank the parameters. Figure 49 demonstrates the first 20 parameters with the highest average importance when the output was the grids' oil saturation in the grid-based dataset in January 2008. These subsets can be determined by having the gradient of importance shown in this figure. Wherever we have a big gradient of importance, the parameters prior to that point can be selected as a subset. Accordingly, four reduced input vectors containing 6, 8, 17, and 22 parameters out of a total of 41 parameters were selected to train the underlying models with ANN.

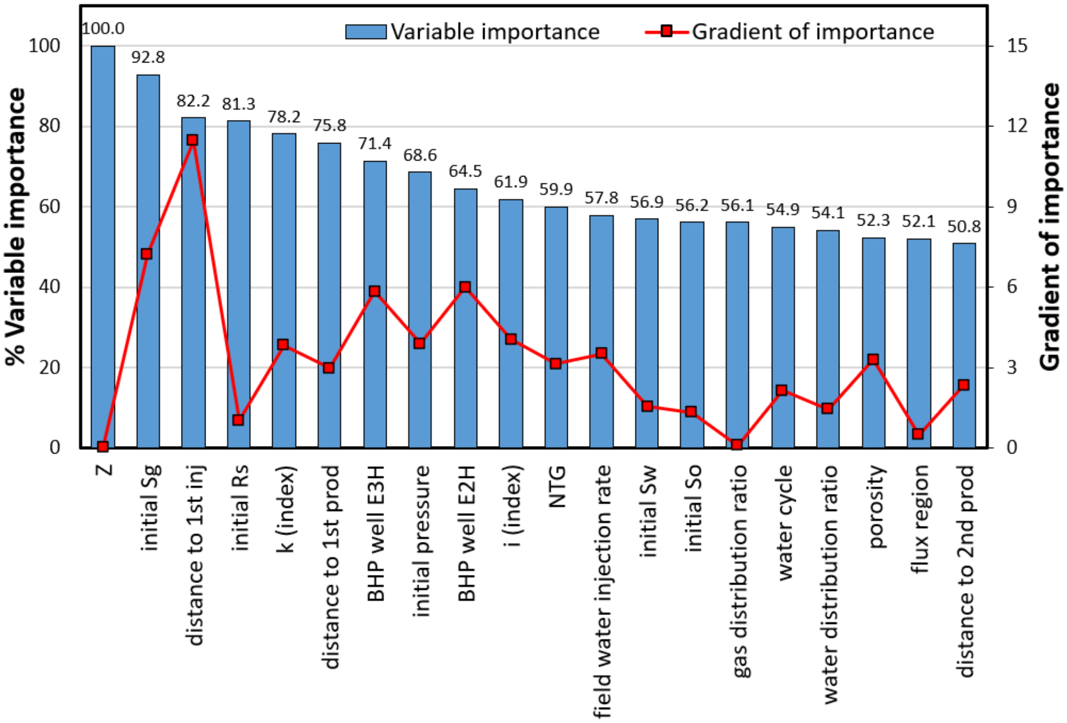


Figure 49. Parameter importance on grids' oil saturation in January 2008

Table 24 displays the goodness of fit for these subsets and two additional subsets of only designing parameters and all 41 parameters. Finally, the subset with 17 parameters was selected to check if the 1D-CNN model could further improve the predictions. We did not have the design parameters related to the gas slugs among the selected parameters. This means gas injection rates were less influential on the grids' oil saturation, and it is possible to predict the oil saturation even without considering them in the model.

Table 24. ANN models' goodness of fit at different subsets for grid-based dataset (Jan 2008)

Test	Selected subset	Test set	
		R^2 (%)	RMSE
1	LHS design parameters + initial Sg + Z	54.42	0.1424
2	First 6 parameters from feature selection	76.83	0.1015
3	First 8 parameters from feature selection	80.34	0.0934
4	First 17 parameters from feature selection	94.26	0.0505
5	First 22 parameters from feature selection	94.73	0.0484
6	All parameters	94.97	0.0473

The 1D-CNN model was also trained on the same dataset, including 17 parameters, to compare it to the ANN model. The best results obtained in a 1D-CNN model consist of three convolutional layers (first and second layers, each containing 100 filters, third layer containing 70 filters and kernel size equal to four) and two fully connected layers (each containing 100 neurons). The goodness of fit for the 1D-CNN model was improved compared to the ANN model (R^2 = 99.38, and RMSE=0.0374). Other CNN models for various timesteps also resulted in R^2 of around 99%. To validate the trained models, a blind test presented in Table 25 was randomly created, and R^2 was achieved at 94-99%. In the initial design with 20 sample points, R^2 was obtained at 85-87%, and five more sample points were added using sequential sampling to have the R^2 of 94-99% for different timesteps.

Table 25. Blind test for WAG scenario

Parameter	Value	Unit
Gas cycle	13.7	month
Water cycle	6.7	month
Gas injection rate (well F-1H)	1.67	MMSCM/d
Gas injection rate (well F-3H)	2.77	MMSCM/d
Water injection rate (well F-1H)	1.16	MSCM/d
Water injection rate (well F-3H)	21.6	MSCM/d
BHP (well E-2H)	188	bar
BHP (well E-3H)	247	bar
BHP (well E-3AH)	233	bar

5.3.4. Well-Based Underlying Model

We performed a similar procedure to train and validate the underlying model for the well-based dataset. Figure 50 demonstrates the first 20 parameters with the highest average importance when the output was cumulative oil production. Subsequently, four reduced input vectors containing 4, 8, 17, and 19 parameters out of a total of 41 parameters were selected to train the underlying models with ANN.

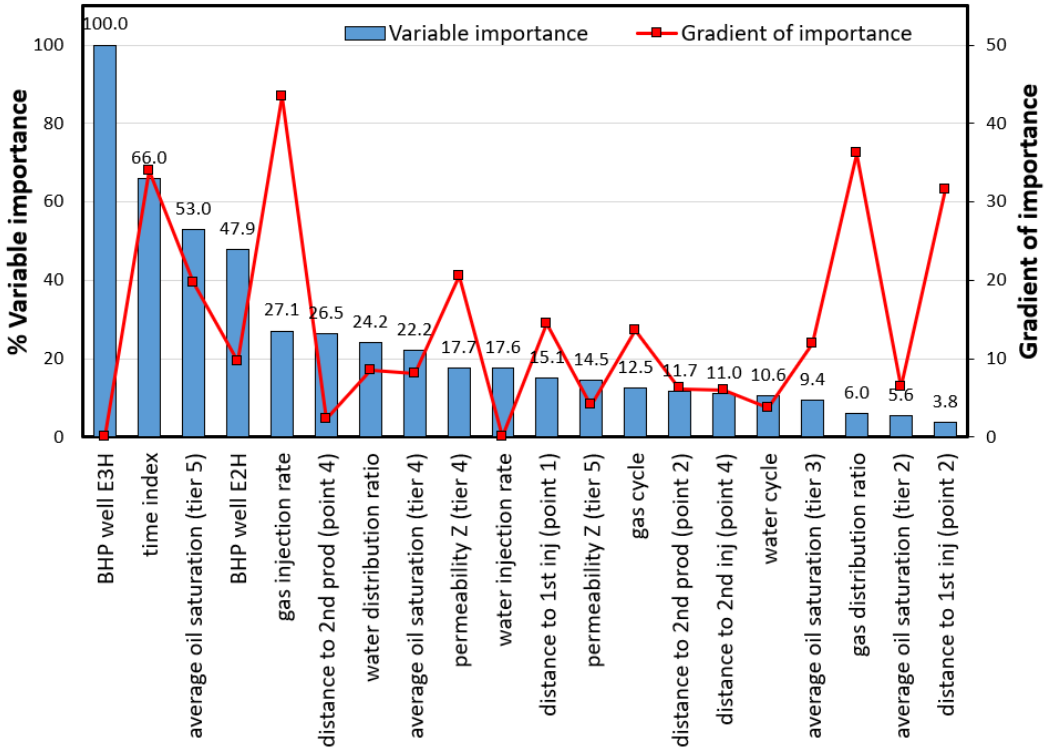


Figure 50. Parameter importance and gradient of importance on cumulative oil production

Table 26. ANN models’ performance at different subsets for the well-based dataset

Test	Selected subset	Test set		
		R^2 (%)	RMSE (Sm^3)	MAPE (%)
1	LHS design parameters + index parameters	96.57	2.8E5	2.61
2	First 4 parameters from feature selection	26.44	15.6E5	10.30
3	First 8 parameters from feature selection	90.25	4.43E5	2.97
4	First 17 parameters from feature selection	98.81	2.29E5	1.65
5	First 19 parameters from feature selection	99.94	0.96E5	0.81
6	All 57 parameters	98.13	1.67E5	1.22

Table 26 presents the ANN models’ performance at different subsets for the well-based dataset. We selected the subset containing 19 parameters and tried to improve the performance using the 1D-CNN on the well-based dataset; however, it did not outperform the ANN model. To validate the trained model, a blind test presented in Table 25 was used, and R^2, RMSE, and MAPE were obtained at 99.80, 1.18E+5, and 1.6, respectively.

5.3.5. Screening and Production Forecasting

The grid-based SPMs were used to screen the WAG process at different timesteps. The target parameters for screening in this study were the grid blocks' oil, water, and gas saturations. The results obtained from the blind test at layer one and timestep one (May 2005) are presented in Figure 51, and more figures corresponding to other layers and timesteps can be found in Appendix B. These figures demonstrate the predicted values from the grid-based SPMs, the simulated values (i.e., actual values), and the error between them. In WAG optimization, it is important to screen the water and gas propagation in the reservoir. The generated figures from grid-based SPM take less than a second to demonstrate the fluids' saturations at different reservoir layers.

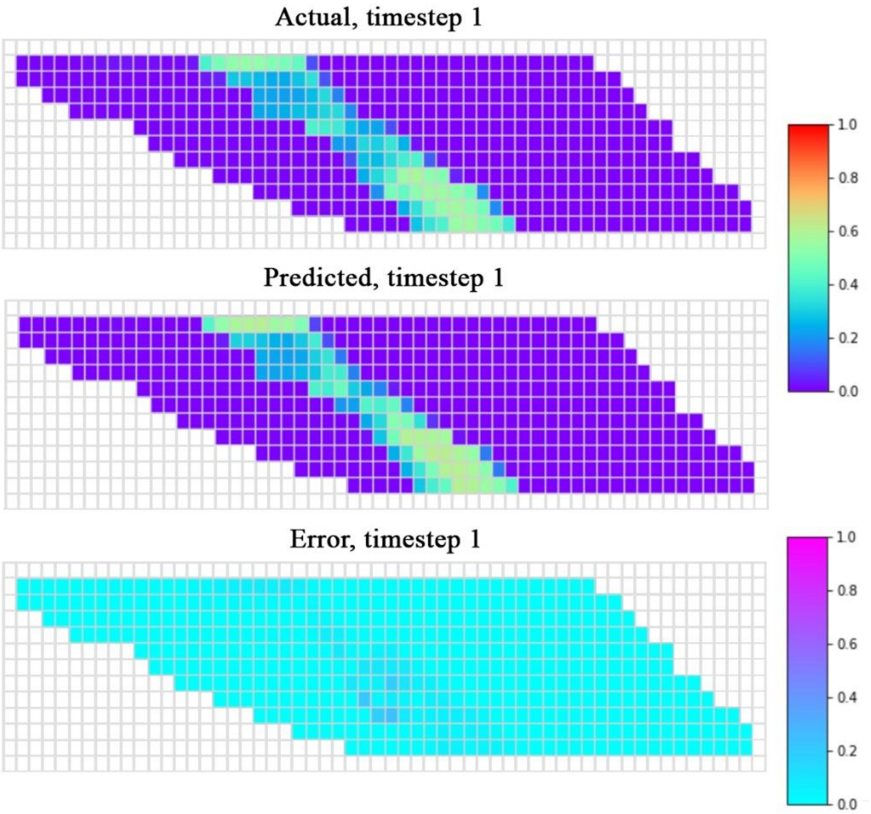


Figure 51. Grids oil saturation in L1, and T1 (May 2005) for the blind test (WAG case)

The well-based SPM was implemented to predict the cumulative oil production for the individual wells at different timesteps at the same blind test. The field's cumulative oil production can then be calculated. Figure 52 shows these predicted and calculated results. It is evident that the coefficients of determination R^2 for wells E-3H and E-3AH were not promising. The production for these two wells quickly dropped to zero when the BHP fell below their respective set pressures of 247 and 233 bar. This behavior of zero production was observed only in two runs, indicating that the model did not have enough data to be trained and to predict this outcome. In order to address this issue, it was necessary to include more sample points to improve the R^2 value. However, in this particular case, we did not add more sample points. The reason for this decision was that the mean absolute percentage error (MAPE) for these two wells was less than 3 percent, and the overall field production was predominantly influenced by the production of well

E-2H. It is evident from the field production data that the prediction accuracy and goodness of fit are highly reliable.

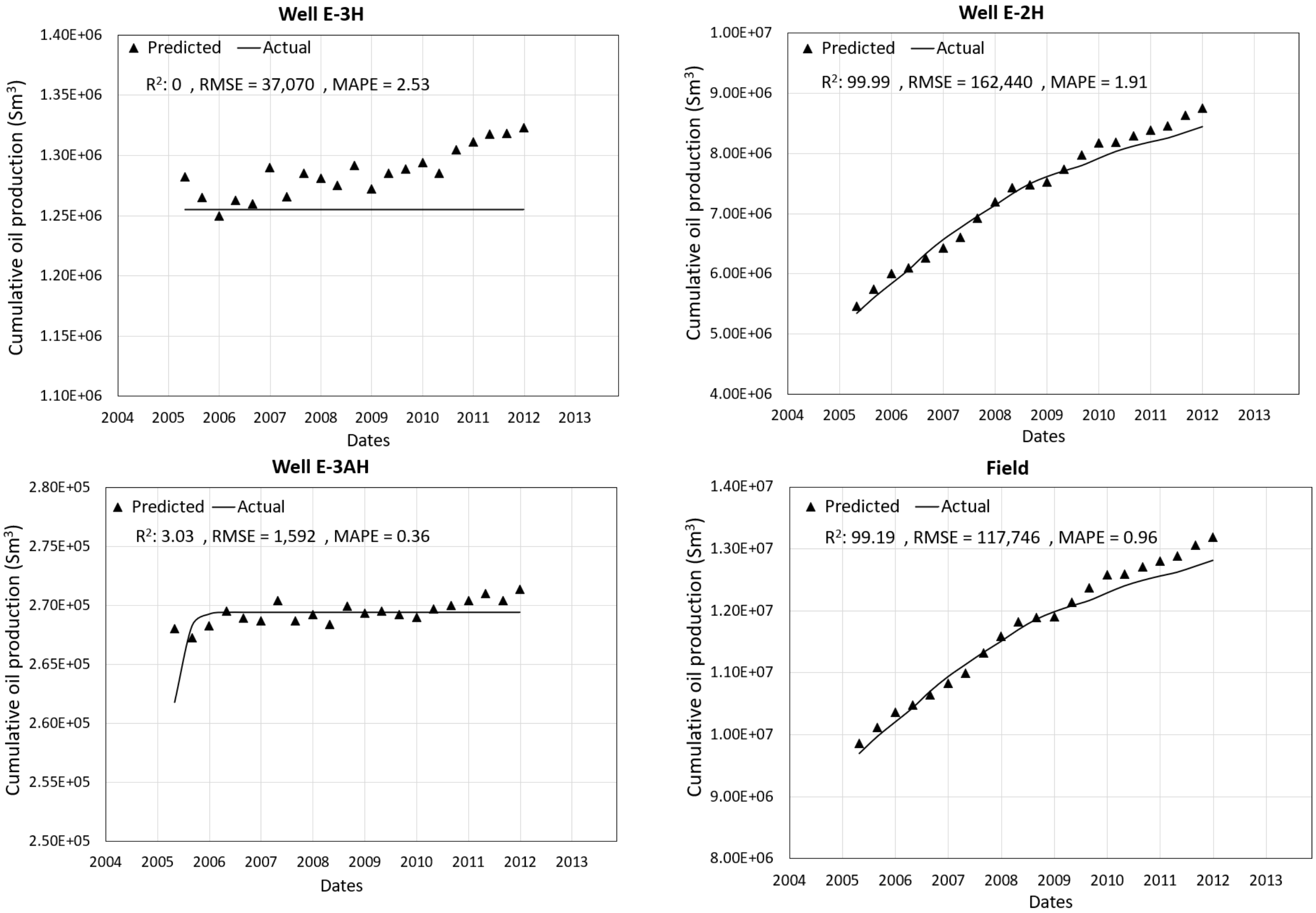


Figure 52. Blind test predicted cumulative oil production of the individual wells, and the calculated results (from well predictions) for the field

5.3.6. Optimization

This case study aimed to maximize the field's cumulative oil production at the last time step. The fitness function in this optimization problem was the well-based SPM. When the well-based SPM takes an input vector of the WAG design parameters, it predicts the individual well's cumulative oil production at the desired timestep. Then, the summation of production for all the producers can be calculated and optimized.

As discussed in section 4.5.5, to avoid the possibility that SPM extrapolates the result, the ranges of the design parameters were shortened to the values presented in Table 27.

Table 27. New ranges for the WAG design parameters in the optimization problem

Parameter	Range	Unit
Gas cycle	4-23	months
Water cycle	4-23	months
Field gas injection rate	3.3-8.7	MMSCM/d
Field water injection rate	10.5-31.5	MSCM/d
Gas distribution ratio	0.1-0.9	
Water distribution ratio	0.1-0.9	
BHP (Well E-2H)	105-255	bar
BHP (Well E-3H)	105-255	bar
BHP (Well E-3AH)	105-235	bar

For the PSO, we selected 200 particles to be distributed randomly in the design parameters' domain space. The number of iterations was equal to 40, inertia was equal to 0.65, and acceleration coefficients c_1 and c_2 were kept constant at 0.8 and 1.4. Consequently, the total number of fitness function evaluations was 200 for one iteration and 8,000 for all iterations. In this work, five realizations based on random particles' positions were introduced. Table 28 presents the best particle values and the global optimum at different PSO realizations. Some other calculated parameters are also listed in this table. Figure 53 displays the cumulative oil production versus iterations at PSO realizations.

Table 28. Summary of random PSO realizations in WAG optimization

	Design Parameter	Unit	Realization 1	Realization 2	Realization 3	Realization 4	Realization 5
Optimization parameters	Gas cycle	month	4	5.8	13.5	5.8	13.5
	Water cycle	month	4	5.8	15.1	5.8	15.1
	Field gas inj rate	MMSCM/d	8.7	7.31	5	7.31	5
	Field water inj rate	MSCM/d	31.5	29.9	29.2	29.9	29.2
	Gas distribution ratio		0.21	0.23	0.9	0.23	0.9
	Water distribution ratio		0.87	0.9	0.34	0.9	0.34
	BHP (well E-2H)	bar	105	105	105	105	105
	BHP (well E-3H)	bar	242	245	219	245	219
	BHP (well E-3AH)	bar	164	121	126	121	126
Calculated	Gas inj rate (well F-1H)	MMSCM/d	1.83	1.68	4.5	1.68	4.5
	Gas inj rate (well F-3H)	MMSCM/d	6.87	5.62	0.5	5.62	0.5
	Water inj rate (well F-1H)	MSCM/d	27.4	26.9	9.9	26.9	9.9
	Water inj rate (well F-3H)	MSCM/d	4.1	3	19.3	3	19.3
	WAG ratio (well F-1H)		2.65	2.85	0.44	2.85	0.44
	WAG ratio (well F-3H)		0.11	0.09	7.63	0.09	7.63

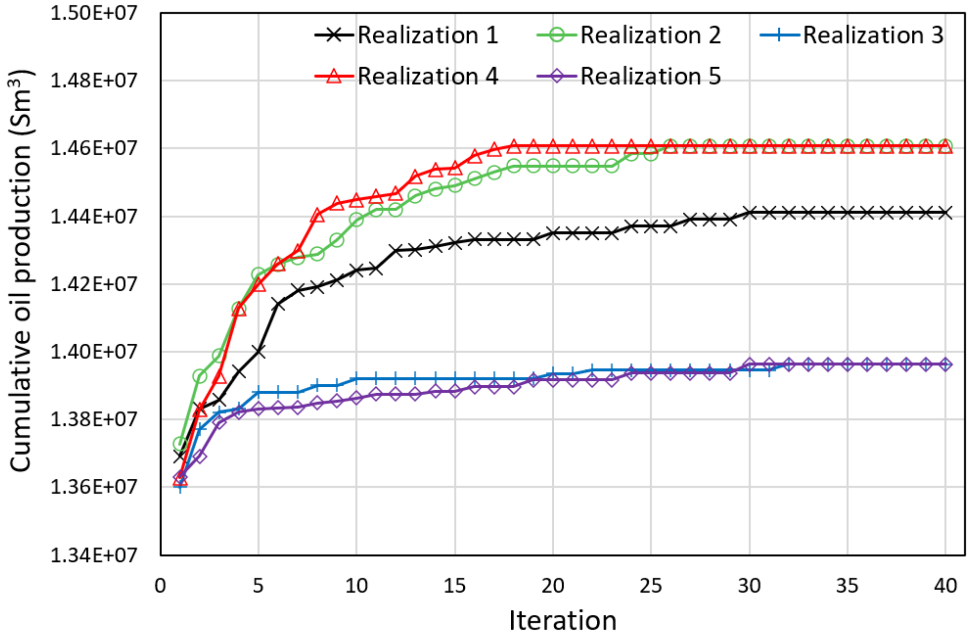


Figure 53. Cumulative oil production vs. iteration at different PSO realizations in WAG optimization

Based on the results, PSO found the global optimum at realizations 2 and 4. PSO converged the results to the global optimum in fewer than 30 iterations. Furthermore, in five realizations of PSO, three realizations failed and were trapped in a local optimum. Optimum parameters in realizations 2 and 4 (Table 28) had smaller WAG cycles. The WAG ratio is generally selected between 0.5 and 4 in most of the cases, but lower values are also observed in literature [236]. Low values of WAG ratio ensure that a substantial gas slug is injected before water breakthrough, but it should be nonzero to maintain macroscopic sweep efficiency. For this case study, the WAG ratio for well F-1H in realizations 2 and 4 was 2.85; however, it was 0.09 for well F-3H which was smaller than the normal range of WAG ratio. Furthermore, field water and gas injection rates were closer to the higher ranges, and for the water cycles, most of the water was injected through well F-1H. Calculated WAG ratio in this table for realization 2 and 4 could also lead to higher recovery. The next highest converged cumulative oil production was related to realization one, in which the optimum parameters had very close values to realization 2 and 4. A final validation was performed to check the result of the numerical model at the optimum parameters in realizations 2 and 4. The field cumulative oil production in the last timestep was obtained at 1.453E+7 Sm^3 compared to the 1.461E+7 Sm^3 recorded from the well-based SPM. This difference shows that the SPM tended to slightly over-predict for the cumulative oil production.

5.4. Conclusions

In this chapter, we developed both grid-based SPMs and grid-based SPM to quickly reproduce the results from the reservoir numerical model to screen and optimize the production in the Norne E-segment WAG case study. The same procedure introduced in chapters 3 and 4 was followed to construct the SPMs. The 1D-CNN underlying model for the grid-based dataset performed better than the ANN model. However, 1D-CNN could not increase the accuracy associated with the well-based dataset. It can be concluded that the 1D-CNN model is best fit into the big datasets (such as the grid-based dataset in this work)

The sampling in this work started with 20 sample points distributed in the domain space of nine design parameters related to the WAG process. The trained CNN models gave an accuracy of 85-87% for different timesteps of the grid-based dataset at the blind test. However, after adding five more sample points using the sequential LHS, the accuracy increased to 94-99%. The R^2, RMSE, and MAPE were obtained for the well-based dataset at 99.80, 1.18E+5, and 1.6 at the blind test using the ANN model.

The trained grid-based SPMs successfully predicted the grids' fluid saturations at different reservoir layers. Moreover, the well-based SPM was coupled with PSO to optimize the cumulative oil production. The results showed that the smaller WAG cycles, water and gas injection rates closer to the higher selected ranges, and injecting the water with a higher portion through well F-1H caused the highest oil recovery.

Chapter 6. Conclusions and Future Research

6.1. Conclusions

Engineers mainly use numerical reservoir models to predict reservoir behavior during the reservoir's lifetime. However, making a proper decision is not an easy task. A conventional reservoir contains many complexities with respect to geological characterization and fluid flow behavior. Furthermore, there are always constraints associated with production and injection operations, making it more time-consuming and complex to analyze.

In this research, an approximation of a numerical reservoir model, called SPM, was developed to mitigate the problems associated with the expensive computational cost of simulators. The methodology to construct the SPM was improved at different steps of sampling, feature ranking, underlying model training, and feature engineering steps. The construction of the SPM is an objective-oriented problem, and each SPM targets to predict a specific output. We tested the proposed SPM methodology using two different case studies to examine the performance in different geological complexities.

In the first case, the Volve numerical model was considered to investigate a waterflooding scenario. The objective was to screen and optimize the oil production to have an acceptable picture of the field performance. For screening purposes, grid-based SPMs, and for production optimization, a well-based SPM were constructed. The liquid production rates of the five producers were selected as the design parameters to have a good case for optimization. In the sampling step, we proposed a sequential sampling technique to avoid repeating the construction from the start. The initial sampling was performed by LHS with a GA, and the extra points were added by the optimized augmentation algorithm. After running the simulator at the points, various static and dynamic parameters were extracted and named as feature engineering.

To construct the grid-based SPM, these parameters were the grids' characteristics, location of the grids, distances to the boundary and closest offset well, production data, bottom-hole pressure (BHP), etc. For the well-based case, the parameters were the well types, distances to the closest producers and injectors, production data, and indexing parameters. Also, a tiering system surrounding each well was generated to consider the effect of dynamic parameters related to the grids (such as fluid saturation and pressure) in that tier. The process of getting the dynamic parameters of the grids requires implementing the constructed grid-based SPMs in parallel with the well-based SPM. The model was then named hybrid well-based SPM in this work.

After forming the grid-based and well-based datasets, an average of 23 ranking algorithms was used. The average technique showed that results differed in magnitude and order compared to using only one technique. In the last step of the SPM construction, CNN and ANN underlying models were trained and tested. For the grid-based SPM, CNN models outperformed the ANN and showed they were better candidates to handle big datasets. Nevertheless, using CNN did not have a significant advantage over ANN for the well-based dataset.

The grid-based SPM showed perfect matches in the screening process with the results obtained from the numerical model. We coupled the well-based SPM with two different optimizing algorithms of PSO and GA to find the best selection of designing parameters (individual well's LPR) and to maximize the cumulative oil production. Both optimizers were quite successful in finding the global optimum. However, PSO showed a more reliable and faster convergence to the solution.

The Norne E-segment black oil numerical model was used for the second case study to investigate a WAG scenario. The objective was to test the methodology in another case with different levels of complexities and involving more design parameters. Nine parameters of gas/water injection cycle, field gas/water injection rate, gas/water injection distribution between two injectors, and BHP for production wells E-2H, E3H, and E-3AH were considered as the design parameters. A similar methodology to the first case study was used to get the final grid-based and well-based SPMs. The outputs of the grid-based SPMs this time were oil, water, and gas saturations. The results of grid-based SPM were generated in only a few seconds, and they could perfectly show the water and gas propagation within the reservoir. Moreover, the results obtained from the optimization using the well-based SPM indicated that the reservoir performed better in the case of smaller WAG cycles, higher water and gas injection rates, and injecting the majority of water through well F-1H.

The main benefit of using the SPM is to provide a fast approximation of the numerical model to predict a selected output parameter. The results obtained from the Volve case study showed that SPM could finish the task of 8,000 runs in 11 hours, but this would take years if a numerical model was used. The difference in time would be more significant if a bigger numerical model, with a higher number of wells or a numerical compositional model, were the cases under study.

6.2. Future Perspectives

There are a few recommendations for future research on the subject of SPM:

1. In this work, individual grid-based SPMs were trained for different timesteps to predict the properties of the grids (such as saturations). One way to make a more comprehensive prediction is to train one SPM for all the timesteps. As a result, to predict the grids properties for all timesteps, the output of the first timestep is considered as the inputs for the next timestep, and so on. However, this procedure has a big challenge. Each timestep's prediction is associated with some error, which propagates into the next timesteps, making the predictions poor as we go forward in timesteps. One solution to avoid this error propagation may be using machine learning techniques for time series, such as Long Short-Term Memory (LSTM) which needs to be tested for future research.
2. In this work, a wide variety of static and dynamic parameters was extracted from the reservoir simulator, and we could reach a reasonable level of accuracy for the constructed SPMs. However, there are still more parameters that can be extracted directly from the simulator or can be generated by feature engineering. We can mention different realizations of the geological parameters such as porosity and permeability distribution. The parameters used in this work may not work well for more complex reservoir models. As a result, more parameters can be incorporated into the SPM construction.
3. We constructed the SPM to approximate the black oil simulators in this work. However, the application of SPM can be tested on compositional simulators for cases such as miscible gas injection. Compositional simulators are usually very time-consuming to run, and the usage of SPM becomes bold. Also, the modified steps such as sequential sampling, average feature ranking, and the CNN underlying model or any other types of

machine learning methods can be addressed in the generated SPMs for compositional simulators.

4. It is helpful to investigate the efficiency of constructing the SPM. This can happen by analyzing the time we spent on using different steps such as sequential sampling, using CNN models, and feature engineering, and comparing the accuracy we obtain if we do not use these techniques.

References

[1] M. Norouzi, H. Panjalizadeh, F. Rashidi, and M. R. Mahdiani, “DPR polymer gel treatment in oil reservoirs: A workflow for treatment optimization using static proxy models,” *Journal of Petroleum Science and Engineering*, vol. 153, pp. 97–110, May 2017, doi: 10.1016/j.petrol.2017.03.018.

[2] W. J. Al-Mudhafar, “Polynomial and nonparametric regressions for efficient predictive proxy metamodeling: Application through the CO2-EOR in shale oil reservoirs,” *Journal of Natural Gas Science and Engineering*, vol. 72, 2019, doi: 10.1016/j.jngse.2019.103038.

[3] C. Y. Peng and R. Gupta, “Experimental design in deterministic modelling: Assessing significant uncertainties,” presented at the SPE Asia Pacific Oil and Gas Conference and Exhibition, Jakarta, Indonesia: OnePetro, Sep. 2003. doi: 10.2118/80537-MS.

[4] D. I. Zubarev, “Pros and cons of applying proxy-models as a substitute for full reservoir simulations,” presented at the SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana: OnePetro, Oct. 2009. doi: 10.2118/124815-MS.

[5] L. M. D. Silva, G. D. Avansi, and D. J. Schiozer, “Development of proxy models for petroleum reservoir simulation: a systematic literature review and state-of-the-art,” vol. 7, no. 10, 2020.

[6] S. D. Mohaghegh, “Quantifying Uncertainties Associated With Reservoir Simulation Studies Using Surrogate Reservoir Models,” Paper presented at the SPE Annual Technical Conference and Exhibition, San Antonio, Texas, USA, September 2006. doi: https://doi.org/10.2118/102492-MS

[7] S. D. Mohaghegh, *Data-driven analytics for the geological storage of CO2*. CRC Press, Taylor & Francis Group, 2018. doi: https://doi.org/10.1201/b21913.

[8] V. Gholami, S. D. Mohaghegh, and M. Maysami, “Smart proxy modeling of SACROC CO2-EOR,” *Fluids*, vol. 4, no. 2, 85, Art. no. 2, 2019, doi: 10.3390/fluids4020085.

[9] Q. He, S. D. Mohaghegh, and Z. Liu, “Reservoir simulation using smart proxy in SACROC unit - Case study,” presented at the SPE Eastern Regional Meeting, Canton, Ohio, USA: OnePetro, Sep. 2016. doi: 10.2118/184069-MS.

[10] Y. Haghshenas, M. Emami Niri, S. Amini, and R. Amiri Kolajoobi, “Developing grid-based smart proxy model to evaluate various water flooding injection scenarios,” *Petroleum Science and Technology*, vol. 38, no. 17, pp. 870–881, 2020, doi: 10.1080/10916466.2020.1796703.

[11] S. Amini and S. Mohaghegh, “Application of machine learning and artificial intelligence in proxy modeling for fluid flow in porous media,” *Fluids*, vol. 4, no. 3, Art. no. 3, 2019, doi: 10.3390/fluids4030126.

[12] A. Shahkarami, S. D. Mohaghegh, V. Gholami, and S. A. Haghighat, “Artificial Intelligence (AI) assisted history matching,” presented at the SPE Western North American and Rocky Mountain Joint Meeting, Denver, Colorado: OnePetro, Apr. 2014. doi: 10.2118/169507-MS.

[13] C. S. W. Ng, A. Jahanbani Ghahfarokhi, M. Nait Amar, and O. Torsæter, “Smart proxy modeling of a fractured reservoir model for production optimization: implementation of metaheuristic algorithm and probabilistic application,” *Nat Resour Res*, vol. 30, no. 3, pp. 2431–2462, Jun. 2021, doi: 10.1007/s11053-021-09844-2.

[14] M. Ahmed and N. Qin, “Surrogate-based aerodynamic design optimization: Use of surrogates in aerodynamic design optimization,” presented at the International Conference

on Aerospace Sciences and Aviation Technology, May 2009, pp. 1–26. doi: 10.21608/ASAT.2009.23442.

[15] M. Eldred and D. Dunlavy, "Formulations for surrogate-based optimization with data fit, multifidelity, and reduced-order models," in *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, in Multidisciplinary Analysis Optimization Conferences. American Institute of Aeronautics and Astronautics, 2006. doi: 10.2514/6.2006-7117.

[16] H. Panjalizadeh, N. Alizadeh, and H. Mashhadi, "A workflow for risk analysis and optimization of steam flooding scenario using static and dynamic proxy models," *Journal of Petroleum Science and Engineering*, vol. 121, pp. 78–86, 2014, doi: 10.1016/j.petrol.2014.06.010.

[17] T. Bartz-Beielstein and M. Zaefferer, "Model-based methods for continuous and discrete global optimization," *Applied Soft Computing*, vol. 55, pp. 154–167, 2017, doi: 10.1016/j.asoc.2017.01.039.

[18] R. R. Barton and M. Meckesheimer, "Chapter 18 - Metamodel-based simulation optimization," in *Handbooks in Operations Research and Management Science*, S. G. Henderson and B. L. Nelson, Eds., in Simulation, vol. 13. Elsevier, 2006, pp. 535–574. doi: 10.1016/S0927-0507(06)13018-2.

[19] A. K. Jaber, S. N. Al-Jawad, and A. K. Alhuraishawy, "A review of proxy modeling applications in numerical reservoir simulation," *Arab J Geosci*, vol. 12, no. 22, p. 701, 2019, doi: 10.1007/s12517-019-4891-1.

[20] R. Yondo, E. Andrés, and E. Valero, "A review on design of experiments and surrogate models in aircraft real-time and many-query aerodynamic analyses," *Progress in Aerospace Sciences*, vol. 96, pp. 23–61, 2018, doi: 10.1016/j.paerosci.2017.11.003.

[21] M. Giselle Fernández-Godino, C. Park, N. H. Kim, and R. T. Haftka, "Issues in deciding whether to use multifidelity surrogates," *AIAA Journal*, vol. 57, no. 5, pp. 2039–2054, 2019, doi: 10.2514/1.J057750.

[22] A. Thenon, V. Gervais, and M. L. Ravalec, "Multi-fidelity meta-modeling for reservoir engineering - Application to history matching," *Comput Geosci*, vol. 20, no. 6, pp. 1231–1250, 2016, doi: 10.1007/s10596-016-9587-y.

[23] M. Larson, "Numerical modeling," in *Encyclopedia of Coastal Science*, M. L. Schwartz, Ed., Dordrecht: Springer Netherlands, 2005, pp. 730–733. doi: 10.1007/1-4020-3880-1_232.

[24] J. S. A. D. Carmo, "Physical modelling vs. numerical modelling: Complementarity and learning," *Preprints*, 2020, doi: 10.20944/preprints202007.0753.v1.

[25] J. H. Ferziger and M. Perić, "Introduction to numerical methods," in *Computational Methods for Fluid Dynamics*, J. H. Ferziger and M. Perić, Eds., Berlin, Heidelberg: Springer, 2002, pp. 21–37. doi: 10.1007/978-3-642-56026-2_2.

[26] S. Koziel and L. Leifsson, Eds., *Surrogate-based modeling and optimization: applications in engineering*. New York: Springer-Verlag, 2013. doi: 10.1007/978-1-4614-7551-4.

[27] G. Avansi, V. Rios, and D. Schiozer, "Numerical tuning in reservoir simulation: It is worth the effort in practical petroleum applications," *J Braz. Soc. Mech. Sci. Eng.*, vol. 41, no. 1, p. 59, 2019, doi: 10.1007/s40430-018-1559-9.

[28] Available online: http://www.webofknowledge.com/ (accessed on 14 August 2021).

[29] A. I. J. Forrester, A. Sóbester, and A. J. Keane, *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, Ltd, 2008. doi: 10.1002/9780470770801.

[30] P. Benner, S. Gugercin, and K. Willcox, “A survey of projection-based model reduction methods for parametric dynamical systems,” *SIAM Rev.*, vol. 57, no. 4, pp. 483–531, 2015, doi: 10.1137/130932715.

[31] S. R. Arridge *et al.*, “Approximation errors and model reduction with an application in optical diffusion tomography,” *Inverse Problems*, vol. 22, no. 1, pp. 175–195, 2006, doi: 10.1088/0266-5611/22/1/010.

[32] A. March and K. Willcox, “Provably convergent multifidelity optimization algorithm not requiring high-fidelity derivatives,” *AIAA Journal*, vol. 50, no. 5, pp. 1079–1089, 2012, doi: 10.2514/1.J051125.

[33] A. Cozad, N. V. Sahinidis, and D. C. Miller, “Learning surrogate models for simulation-based optimization,” *AIChE Journal*, vol. 60, no. 6, pp. 2211–2227, 2014, doi: 10.1002/aic.14418.

[34] L. Sirovich, “Turbulence and the dynamics of coherent structures part I: Coherent structures,” *Quarterly of Applied Mathematics*, vol. 45, no. 3, pp. 561–571, 1987.

[35] M. J. Rewienski, “A trajectory piecewise-linear approach to model order reduction of nonlinear dynamical systems,” PhD dissertation, Massachusetts Institute of Technology, 2003.

[36] S. Chaturantabut and D. C. Sorensen, “Nonlinear model reduction via discrete empirical interpolation,” *SIAM J. Sci. Comput.*, vol. 32, no. 5, pp. 2737–2764, 2010, doi: 10.1137/090766498.

[37] X. Wan, J. F. Pekny, and G. V. Reklaitis, “Simulation-based optimization with surrogate models—Application to supply chain management,” *Computers & Chemical Engineering*, vol. 29, no. 6, pp. 1317–1328, 2005, doi: 10.1016/j.compchemeng.2005.02.018.

[38] D. Amsallem and U. Hetmaniuk, “A posteriori error estimators for linear reduced-order models using Krylov-based integrators,” *International Journal for Numerical Methods in Engineering*, vol. 102, no. 5, pp. 1238–1261, 2015, doi: 10.1002/nme.4753.

[39] S. Rouhani and D. E. Myers, “Problems in space-time kriging of geohydrological data,” *Math Geol*, vol. 22, no. 5, pp. 611–623, 1990, doi: 10.1007/BF00890508.

[40] F. Lazzeri, *Machine learning for time series forecasting with Python*. Wiley, 2020.

[41] S. D. Mohaghegh, F. Abdulla, M. Abdou, R. Gaskari, and M. Maysami, “Smart proxy: an innovative reservoir management tool; case study of a giant mature oilfield in the UAE,” presented at the Abu Dhabi International Petroleum Exhibition and Conference, Abu Dhabi, UAE: OnePetro, Nov. 2015. doi: 10.2118/177829-MS.

[42] S. Amini, S. D. Mohaghegh, R. Gaskari, and G. Bromhal, “Uncertainty analysis of a CO2 sequestration project using surrogate reservoir modeling technique,” presented at the SPE Western Regional Meeting, Bakersfield, California, USA: OnePetro, Mar. 2012. doi: 10.2118/153843-MS.

[43] S. D. Mohaghegh, H. H. Hafez, R. Gaskari, M. Haajizadeh, and M. Kenawy, “Uncertainty analysis of a giant oil field in the Middle East using surrogate reservoir model,” presented at the Abu Dhabi International Petroleum Exhibition and Conference, Abu Dhabi, UAE: OnePetro, Nov. 2006. doi: 10.2118/101474-MS.

[44] S. D. Mohaghegh, S. Amini, V. Gholami, R. Gaskari, and G. Bromhal, “Grid-based surrogate reservoir modeling (SRM) for fast track analysis of numerical reservoir simulation models at the grid block level,” presented at the SPE Western Regional Meeting, Bakersfield, California, USA: OnePetro, Mar. 2012. doi: 10.2118/153844-MS.

[45] C. A. Henao and C. T. Maravelias, "Surrogate-based process synthesis," in *Computer Aided Chemical Engineering*, S. Pierucci and G. B. Ferraris, Eds., in 20 European Symposium on Computer Aided Process Engineering, vol. 28. Elsevier, 2010, pp. 1129–1134. doi: 10.1016/S1570-7946(10)28189-0.

[46] S. Hoops *et al.*, "Chapter 5 - Ordinary differential equations (ODEs) based modeling," in *Computational Immunology*, J. Bassaganya-Riera, Ed., Academic Press, 2016, pp. 63–78. doi: 10.1016/B978-0-12-803697-6.00005-9.

[47] S. Simske, "Chapter 5 - Sensitivity analysis and big system engineering," in *Meta-Analytics*, S. Simske, Ed., Morgan Kaufmann, 2019, pp. 187–201. doi: 10.1016/B978-0-12-814623-1.00005-8.

[48] T. Van Steenkiste, J. van der Herten, I. Couckuyt, and T. Dhaene, "Data-efficient sensitivity analysis with surrogate modeling," in *Uncertainty Modeling for Engineering Applications*, F. Canavero, Ed., in PoliTO Springer Series. Springer, Cham, 2019, pp. 55–69. doi: 10.1007/978-3-030-04870-9_4.

[49] S. Amini, S. D. Mohaghegh, R. Gaskari, and G. S. Bromhal, "Pattern recognition and data-driven analytics for fast and accurate replication of complex numerical reservoir models at the grid block level," presented at the SPE Intelligent Energy Conference & Exhibition, Utrecht, The Netherlands: OnePetro, Apr. 2014. doi: 10.2118/167897-MS.

[50] A. Anand, M. Agrawal, N. Bhatt, and M. Ram, "Chapter 11 - Software patch scheduling policy incorporating functional safety standards," in *Advances in System Reliability Engineering*, M. Ram and J. P. Davim, Eds., Academic Press, 2019, pp. 267–279. doi: 10.1016/B978-0-12-815906-4.00011-7.

[51] Z. Hou, W. Lu, and M. Chen, "Surrogate-based sensitivity analysis and uncertainty analysis for DNAPL-contaminated aquifer remediation," *Journal of Water Resources Planning and Management*, vol. 142, no. 11, p. 04016043, 2016, doi: 10.1061/(ASCE)WR.1943-5452.0000677.

[52] B. Iooss and P. Lemaître, "A review on global sensitivity analysis methods," in *Uncertainty Management in Simulation-Optimization of Complex Systems: Algorithms and Applications*, G. Dellino and C. Meloni, Eds., in Operations Research/Computer Science Interfaces Series. Springer, Boston, MA, 2015, pp. 101–122. doi: 10.1007/978-1-4899-7547-8_5.

[53] A. A. Chaudhry, J. Buchwald, and T. Nagel, "Local and global spatio-temporal sensitivity analysis of thermal consolidation around a point heat source," *International Journal of Rock Mechanics and Mining Sciences*, vol. 139, p. 104662, 2021, doi: 10.1016/j.ijrmms.2021.104662.

[54] A. Saltelli and P. Annoni, "How to avoid a perfunctory sensitivity analysis," *Environmental Modelling & Software*, vol. 25, no. 12, pp. 1508–1517, 2010, doi: 10.1016/j.envsoft.2010.04.012.

[55] M. Ye and M. C. Hill, "Chapter 10 - Global sensitivity analysis for uncertain parameters, models, and scenarios," in *Sensitivity Analysis in Earth Observation Modelling*, G. P. Petropoulos and P. K. Srivastava, Eds., Elsevier, 2017, pp. 177–210. doi: 10.1016/B978-0-12-803011-0.00010-0.

[56] S. Razmyan and F. Hosseinzadeh Lotfi, "An application of Monte-Carlo-based sensitivity analysis on the overlap in discriminant analysis," *Journal of Applied Mathematics*, vol. 2012, no. e315868, 2012, doi: 10.1155/2012/315868.

[57] I. M. Sobol, “Sensitivity estimates for nonlinear mathematical models,” *Mathematical Modelling and Computational Experiments*, vol. 1, no. 4, pp. 407–414, 1993.

[58] M. D. Morris, “Factorial sampling plans for preliminary computational experiments,” *Technometrics*, vol. 33, no. 2, pp. 161–174, 1991, doi: 10.1080/00401706.1991.10484804.

[59] X. Song, J. Zhang, C. Zhan, Y. Xuan, M. Ye, and C. Xu, “Global sensitivity analysis in hydrological modeling: Review of concepts, methods, theoretical framework, and applications,” *Journal of Hydrology*, vol. 523, pp. 739–757, 2015, doi: 10.1016/j.jhydrol.2015.02.013.

[60] A. Bhosekar and M. Ierapetritou, “Advances in surrogate based modeling, feasibility analysis, and optimization: A review,” *Computers & Chemical Engineering*, vol. 108, pp. 250–267, 2018, doi: 10.1016/j.compchemeng.2017.09.017.

[61] C. M. Bishop, *Pattern recognition and machine learning*. in Information Science and Statistics. Springer-Verlag New York, 2006.

[62] K. Crombecq, “Surrogate modeling of computer experiments with sequential experimental design,” PhD dissertation, Ghent University. Faculty of Engineering and Architecture ; University of Antwerp. Faculty of Science, Ghent; Antwerp, Belgium, 2011.

[63] L. Pronzato and W. G. Müller, “Design of computer experiments: space filling and beyond,” *Stat Comput*, vol. 22, no. 3, pp. 681–701, 2012, doi: 10.1007/s11222-011-9242-3.

[64] S. S. Garud, I. A. Karimi, and M. Kraft, “Design of computer experiments: a review,” *Computers & Chemical Engineering*, vol. 106, pp. 71–95, 2017, doi: 10.1016/j.compchemeng.2017.05.010.

[65] Y. Choi, D. Song, S. Yoon, and J. Koo, “Comparison of factorial and Latin hypercube sampling designs for meta-models of building heating and cooling loads,” *Energies*, vol. 14, no. 2, 512, 2021, doi: 10.3390/en14020512.

[66] C. Natoli, “Classical designs: Fractional factorial designs,” Scientific Test and Analysis Techniques Center of Excellence (STAT COE), 2950 Hobson Way, Wright-Patterson AFB, OH 45433, Report-02-2018, 2018.

[67] D. Humbird and Q. Fei, “Chapter 20 - Scale-up considerations for biofuels,” in *Biotechnology for Biofuel Production and Optimization*, C. A. Eckert and C. T. Trinh, Eds., Amsterdam: Elsevier, 2016, pp. 513–537. doi: 10.1016/B978-0-444-63475-7.00020-0.

[68] Z. Hajjar, M. Kazemeini, A. Rashidi, and S. Soltanali, “Optimizing parameters affecting synthesis of a novel Co–Mo/GO catalyst in a Naphtha HDS reaction utilizing D-optimal experimental design method,” *Journal of the Taiwan Institute of Chemical Engineers*, vol. 78, pp. 566–575, 2017, doi: 10.1016/j.jtice.2017.06.048.

[69] M. D. McKay, R. J. Beckman, and W. J. Conover, “Comparison of three methods for selecting values of input variables in the analysis of output from a computer code,” *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979, doi: 10.1080/00401706.1979.10489755.

[70] F. A. C. Viana, G. Venter, and V. Balabanov, “An algorithm for fast optimal Latin hypercube design of experiments,” *International Journal for Numerical Methods in Engineering*, vol. 82, no. 2, pp. 135–156, 2010, doi: 10.1002/nme.2750.

[71] A. S. Hedayat, N. J. A. Sloane, and J. Stufken, *Orthogonal arrays: Theory and applications*. in Springer Series in Statistics. New York: Springer-Verlag, 1999. doi: 10.1007/978-1-4612-1478-6.

[72] J. Eason and S. Cremaschi, “Adaptive sequential sampling for surrogate model generation with artificial neural networks,” *Computers & Chemical Engineering*, vol. 68, pp. 220–232, 2014, doi: 10.1016/j.compchemeng.2014.05.021.

[73] F. Provost, D. Jensen, and T. Oates, “Efficient progressive sampling,” in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, in KDD ’99. New York, NY, USA: Association for Computing Machinery, Aug. 1999, pp. 23–32. doi: 10.1145/312129.312188.

[74] D. Gamerman and H. F. Lopes, *Markov chain Monte Carlo: Stochastic simulation for Bayesian inference*, 2nd edition. Chapman and Hall/CRC, 2006.

[75] I. M. Sobol, “On the distribution of points in a cube and the approximate evaluation of integrals,” *USSR Computational Mathematics and Mathematical Physics*, vol. 7, no. 4, pp. 86–112, 1967, doi: 10.1016/0041-5553(67)90144-9.

[76] J. H. Halton, “On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals,” *Numer. Math.*, vol. 2, no. 1, pp. 84–90, 1960, doi: 10.1007/BF01386213.

[77] J. Cheng and M. J. Druzdzel, “Computational investigation of low-discrepancy sequences in simulation algorithms for bayesian networks,” *arXiv:1301.3841*, 2013.

[78] P. Z. G. Qian, “Nested Latin hypercube designs,” *Biometrika*, vol. 96, no. 4, pp. 957–970, 2009.

[79] G. Rennen, B. Husslage, E. R. Van Dam, and D. Den Hertog, “Nested maximin Latin hypercube designs,” *Struct Multidisc Optim*, vol. 41, no. 3, pp. 371–395, 2010, doi: 10.1007/s00158-009-0432-y.

[80] F. Xiong, Y. Xiong, W. Chen, and S. Yang, “Optimizing Latin hypercube design for sequential sampling of computer experiments,” *Engineering Optimization*, vol. 41, no. 8, pp. 793–810, 2009, doi: 10.1080/03052150902852999.

[81] R. R. Barton, “Metamodels for simulation input-output relations,” in *Proceedings of the 24th Conference on Winter Simulation, WSC 1992*, Institute of Electrical and Electronics Engineers Inc., Dec. 1992, pp. 289–299. doi: 10.1145/167293.167352.

[82] W. C. M. van Beers and J. P. C. Kleijnen, “Kriging interpolation in simulation: A survey,” in *Proceedings of the 2004 Winter Simulation Conference, 2004.*, Dec. 2004, p. 121. doi: 10.1109/WSC.2004.1371308.

[83] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, “Design and analysis of computer experiments,” *Statistical Science*, vol. 4, no. 4, pp. 409–423, 1989, doi: 10.1214/ss/1177012413.

[84] M. J. Sasena, “Flexibility and efficiency enhancements for constrained global design optimization with kriging approximations.,” PhD dissertation, University of Michigan, 2002.

[85] J. Sacks, S. B. Schiller, and W. J. Welch, “Designs for computer experiments,” *Technometrics*, vol. 31, no. 1, pp. 41–47, 1989, doi: 10.2307/1270363.

[86] T. W. Simpson, J. D. Poplinski, P. N. Koch, and J. K. Allen, “Metamodels for computer-based engineering design: Survey and recommendations,” *EWC*, vol. 17, no. 2, pp. 129–150, 2001, doi: 10.1007/PL00007198.

[87] K. McBride and K. Sundmacher, “Overview of surrogate modeling in chemical process engineering,” *Chemie Ingenieur Technik*, vol. 91, no. 3, pp. 228–239, 2019, doi: 10.1002/cite.201800091.

[88] J. H. Friedman, "Multivariate adaptive regression splines," *The Annals of Statistics*, vol. 19, no. 1, pp. 1–67, 1991, doi: 10.1214/aos/1176347963.

[89] E. Quirós, Á. M. Felicísimo, and A. Cuartero, "Testing multivariate adaptive regression splines (MARS) as a method of land cover classification of TERRA-ASTER satellite images," *Sensors*, vol. 9, no. 11, pp. 9011–9028, 2009, doi: 10.3390/s91109011.

[90] W. Zhang and A. T. C. Goh, "Multivariate adaptive regression splines and neural network models for prediction of pile drivability," *Geoscience Frontiers*, vol. 7, no. 1, pp. 45–52, 2016, doi: 10.1016/j.gsf.2014.10.003.

[91] N. Van Nguyen, J.-W. Lee, M. Tyan, and S. Kim, "Repetitively enhanced neural networks method for complex engineering design optimisation problems," *The Aeronautical Journal*, vol. 119, no. 1220, pp. 1253–1270, 2015, doi: 10.1017/S0001924000011234.

[92] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: comparison of trends in practice and research for deep learning," *arXiv:1811.03378 [cs]*, 2018, [Online]. Available: http://arxiv.org/abs/1811.03378

[93] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, p. e00938, Nov. 2018, doi: 10.1016/j.heliyon.2018.e00938.

[94] S.-C. Huang and T.-H. Le, "Chapter 8 - Convolutional neural network architectures," in *Principles and Labs for Deep Learning*, S.-C. Huang and T.-H. Le, Eds., Academic Press, 2021, pp. 201–217. doi: 10.1016/B978-0-323-90198-7.00001-X.

[95] S. O. Haykin, *Neural networks and learning machines, 3rd edition*, 3rd ed. Pearson, 2009.

[96] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 302–309, 1991, doi: 10.1109/72.80341.

[97] V. Vapnik, *The nature of statistical learning theory*, 2nd ed. in Information Science and Statistics. New York: Springer-Verlag New York, 2000. doi: 10.1007/978-1-4757-3264-1.

[98] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in Neural Information Processing Systems*, M. C. Mozer, M. Jordan, and T. Petsche, Eds., MIT Press, 1997.

[99] M. Awad and R. Khanna, "Support vector regression," in *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, M. Awad and R. Khanna, Eds., Berkeley, CA: Apress, 2015, pp. 67–80. doi: 10.1007/978-1-4302-5990-9_4.

[100] J. R. Koza, "Genetic programming as a means for programming computers by natural selection," *Stat Comput*, vol. 4, no. 2, pp. 87–112, 1994, doi: 10.1007/BF00175355.

[101] Y. Liao, J. Wang, B. Meng, and X. Li, "Integration of GP and GA for mapping population distribution," *International Journal of Geographical Information Science*, vol. 24, no. 1, pp. 47–67, 2010, doi: 10.1080/13658810802186874.

[102] C. Raymond, Q. Chen, B. Xue, and M. Zhang, "Genetic programming with rademacher complexity for symbolic regression," in *2019 IEEE Congress on Evolutionary Computation (CEC)*, 2019, pp. 2657–2664. doi: 10.1109/CEC.2019.8790341.

[103] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[104] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998, doi: 10.1109/34.709601.

[105] C. B. Storlie, L. P. Swiler, J. C. Helton, and C. J. Sallaberry, “Implementation and evaluation of nonparametric regression procedures for sensitivity analysis of computationally demanding models,” *Reliability Engineering & System Safety*, vol. 94, no. 11, pp. 1735–1763, 2009, doi: 10.1016/j.ress.2009.05.007.

[106] T. Chen and C. Guestrin, “XGBoost: a scalable tree boosting system,” *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016, doi: 10.1145/2939672.2939785.

[107] J. H. Friedman, “Greedy function approximation: a gradient boosting machine.,” *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001, doi: 10.1214/aos/1013203451.

[108] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, “A comparative analysis of XGBoost,” *Artif Intell Rev*, vol. 54, no. 3, pp. 1937–1967, 2021, doi: 10.1007/s10462-020-09896-5.

[109] R. Zhong, R. Johnson, and Z. Chen, “Generating pseudo density log from drilling and logging-while-drilling data using extreme gradient boosting (XGBoost),” *International Journal of Coal Geology*, vol. 220, 2020, doi: 10.1016/j.coal.2020.103416.

[110] N. Wiener, “The homogeneous chaos,” *American Journal of Mathematics*, vol. 60, no. 4, pp. 897–936, 1938, doi: 10.2307/2371268.

[111] Y. Zhou, Z. Lu, J. Hu, and Y. Hu, “Surrogate modeling of high-dimensional problems via data-driven polynomial chaos expansions and sparse partial least square,” *Computer Methods in Applied Mechanics and Engineering*, vol. 364, 2020, doi: 10.1016/j.cma.2020.112906.

[112] D. Xiu and G. E. Karniadakis, “The Wiener--Askey polynomial chaos for stochastic differential equations,” *SIAM J. Sci. Comput.*, vol. 24, no. 2, pp. 619–644, 2002, doi: 10.1137/S1064827501387826.

[113] T. Jain, R. G. Patel, and J. Trivedi, “Application of polynomial chaos theory as an accurate and computationally efficient proxy model for heterogeneous steam-assisted gravity drainage reservoirs,” *Energy Science & Engineering*, vol. 5, no. 5, pp. 270–289, 2017, doi: 10.1002/ese3.177.

[114] A. Ebrahimi and E. Khamehchi, “Developing a novel workflow for natural gas lift optimization using advanced support vector machine,” *Journal of Natural Gas Science and Engineering*, vol. 28, pp. 626–638, 2016, doi: 10.1016/j.jngse.2015.12.031.

[115] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. in Complex Adaptive Systems. Cambridge, MA, USA: A Bradford Book, 1992.

[116] O. Erdinc and M. Uzunoglu, “Optimum design of hybrid renewable energy systems: Overview of different approaches,” *Renewable and Sustainable Energy Reviews*, vol. 16, no. 3, pp. 1412–1425, 2012, doi: 10.1016/j.rser.2011.11.011.

[117] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, Nov. 1995, pp. 1942–1948 vol.4. doi: 10.1109/ICNN.1995.488968.

[118] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” in *Readings in Computer Vision*, M. A. Fischler and O. Firschein, Eds., San Francisco (CA): Morgan Kaufmann, 1987, pp. 606–615. doi: 10.1016/B978-0-08-051581-6.50059-3.

[119] H. H. A. Mukhairez and A. Y. A. Maghari, “Performance comparison of simulated annealing, GA and ACO Applied to TSP,” *IJICR*, vol. 6, no. 4, pp. 647–654, 2015, doi: 10.20533/ijicr.2042.4655.2015.0080.

[120] J. Qin, L. Ni, and F. Shi, “Combined simulated annealing algorithm for the discrete facility location problem,” *The Scientific World Journal*, vol. 2012, p. e576392, 2012, doi: 10.1100/2012/576392.

[121] M. Dorigo, V. Maniezzo, and A. Colorni, “Ant system: Optimization by a colony of cooperating agents,” *IEEE Trans Syst Man Cybern B Cybern*, vol. 26, no. 1, pp. 29–41, 1996, doi: 10.1109/3477.484436.

[122] K. Price, R. M. Storn, and J. A. Lampinen, *Differential evolution: a practical approach to global optimization*. in Natural Computing Series. Berlin Heidelberg: Springer-Verlag, 2005. doi: 10.1007/3-540-31306-0.

[123] V. Kachitvichyanukul, “Comparison of three evolutionary algorithms: GA, PSO, and DE,” *Industrial Engineering and Management Systems*, vol. 11, no. 3, pp. 215–223, 2012, doi: 10.7232/iems.2012.11.3.215.

[124] E. P. dos Santos Amorim, C. R. Xavier, R. S. Campos, and R. W. dos Santos, “Comparison between genetic algorithms and differential evolution for solving the history matching problem,” in *Computational Science and Its Applications – ICCSA 2012*, B. Murgante, O. Gervasi, S. Misra, N. Nedjah, A. M. A. C. Rocha, D. Taniar, and B. O. Apduhan, Eds., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2012, pp. 635–648. doi: 10.1007/978-3-642-31125-3_48.

[125] A. Slowik and H. Kwasnicka, “Evolutionary algorithms and their applications to engineering problems,” *Neural Comput & Applic*, vol. 32, no. 16, pp. 12363–12379, 2020, doi: 10.1007/s00521-020-04832-8.

[126] M. Georgioudakis and V. Plevris, “A comparative study of differential evolution variants in constrained structural optimization,” *Frontiers in Built Environment*, vol. 6, p. 102, 2020, doi: 10.3389/fbuil.2020.00102.

[127] R. P. Batycky, M. J. Blunt, and M. R. Thiele, “A 3D field-scale streamline-based reservoir simulator,” *SPE Reservoir Engineering*, vol. 12, no. 4, pp. 246–254, 1997, doi: 10.2118/36726-PA.

[128] S. Tanaka, T. Onishi, D. Kam, K. Dehghani, and X.-H. Wen, “Application of combined streamline based reduced-physics surrogate and response surface method for field development optimization,” presented at the International Petroleum Technology Conference, Dhahran, Kingdom of Saudi Arabia: OnePetro, Jan. 2020. doi: 10.2523/IPTC-19958-MS.

[129] M. R. Thiele and R. P. Batycky, “Water injection optimization using a streamline-based workflow,” presented at the SPE Annual Technical Conference and Exhibition, Denver, Colorado: OnePetro, Oct. 2003. doi: 10.2118/84080-MS.

[130] A. R. Kovscek and Y. Wang, “Geologic storage of carbon dioxide and enhanced oil recovery. I. Uncertainty quantification employing a streamline based proxy for reservoir flow simulation,” *Energy Conversion and Management*, vol. 46, no. 11–12, pp. 1920–1940, 2005, doi: 10.1016/j.enconman.2004.09.008.

[131] Y. Wang and A. R. Kovscek, “Streamline approach for history matching production data,” *SPE Journal*, vol. 5, no. 4, pp. 353–362, 2000, doi: 10.2118/58350-PA.

[132] V. R. Stenerud, V. Kippe, K.-A. Lie, and A. Datta-Gupta, “Adaptive multiscale streamline simulation and inversion for high-resolution geomodels,” *SPE Journal*, vol. 13, no. 1, pp. 99–111, 2008, doi: 10.2118/106228-PA.

[133] D. Kam, J. Han, and A. Datta-Gupta, “Streamline-based history matching of bottomhole pressure and three-phase production data using a multiscale approach,” *Journal of*

Petroleum Science and Engineering, vol. 154, pp. 217–233, 2017, doi: 10.1016/j.petrol.2017.04.022.

[134] W. J. Milliken, A. S. Emanuel, and A. Chakravarty, "Applications of 3D streamline simulation to assist history matching," *SPE Reservoir Evaluation & Engineering*, vol. 4, no. 6, pp. 502–508, 2001, doi: 10.2118/74712-PA.

[135] S. Taware *et al.*, "Well placement optimization in a mature carbonate waterflood using streamline-based quality maps," presented at the SPE Oil and Gas India Conference and Exhibition, Mumbai, India: OnePetro, Mar. 2012. doi: 10.2118/155055-MS.

[136] M. R. Thiele, R. P. Batycky, M. J. Blunt, and F. M. Orr Jr., "Simulating flow in heterogeneous systems using streamtubes and streamlines," *SPE Reservoir Engineering*, vol. 11, no. 1, pp. 5–12, 1996, doi: 10.2118/27834-PA.

[137] A. Datta-Gupta and M. J. King, *Streamline simulation: Theory and practice*. Society of Petroleum Engineers, 2007.

[138] G. Bardy, P. Biver, G. Caumon, and P. Renard, "Oil production uncertainty assessment by predicting reservoir production curves and confidence intervals from arbitrary proxy responses," *Journal of Petroleum Science and Engineering*, vol. 176, pp. 116–125, 2019, doi: 10.1016/j.petrol.2019.01.035.

[139] F. A. Allam, A. H. El-Banbi, S. S. Bustami, T. H. Saada, and I. I. Fahmy, "History match tuning through different upscaling algorithms," presented at the SPE Annual Technical Conference and Exhibition, Houston, Texas: OnePetro, Sep. 2004. doi: 10.2118/90292-MS.

[140] H. Li and L. J. Durlofsky, "Upscaling for compositional reservoir simulation," *SPE Journal*, vol. 21, no. 3, pp. 0873–0887, 2016, doi: 10.2118/173212-PA.

[141] Y. Yang, X. Wang, X.-H. Wu, and L. Bi, "Multiphase upscaling using approximation techniques," presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA: OnePetro, Feb. 2013. doi: 10.2118/163655-MS.

[142] V. S. Rios, L. O. S. Santos, F. B. Quadros, and D. J. Schiozer, "New upscaling technique for compositional reservoir simulations of miscible gas injection," *Journal of Petroleum Science and Engineering*, vol. 175, pp. 389–406, 2019, doi: 10.1016/j.petrol.2018.12.061.

[143] W. A. Bruce, "An electrical device for analyzing oil-reservoir behavior," *Transactions of the AIME*, vol. 151, no. 1, pp. 112–124, 1943, doi: 10.2118/943112-G.

[144] A. A. Yousef, P. H. Gentil, J. L. Jensen, and L. W. Lake, "A capacitance model to infer interwell connectivity from production and injection rate fluctuations," *SPE Reservoir Evaluation & Engineering*, vol. 9, no. 6, pp. 630–646, 2006, doi: 10.2118/95322-PA.

[145] M. Almarri, B. Prakasa, K. Muradov, and D. Davies, "Identification and characterization of thermally induced fractures using modern analytical techniques," presented at the SPE Kingdom of Saudi Arabia Annual Technical Symposium and Exhibition, Dammam, Saudi Arabia: OnePetro, Apr. 2017. doi: 10.2118/188084-MS.

[146] R. W. de Holanda, E. Gildin, J. L. Jensen, L. W. Lake, and C. S. Kabir, "A state-of-the-art literature review on capacitance resistance models for reservoir characterization and performance forecasting," *Energies*, vol. 11, no. 12, 3368, 2018, doi: 10.3390/en11123368.

[147] E. Artun, "Characterizing reservoir connectivity and forecasting waterflood performance using data-driven and reduced-physics models," presented at the SPE Western Regional Meeting, Anchorage, Alaska, USA: OnePetro, May 2016. doi: 10.2118/180488-MS.

[148] K. C. Wilson and L. J. Durlofsky, “Computational optimization of shale resource development using reduced-physics surrogate models,” presented at the SPE Western Regional Meeting, Bakersfield, California, USA: OnePetro, Mar. 2012. doi: 10.2118/152946-MS.

[149] A. Chatterjee, “An introduction to the proper orthogonal decomposition,” *Current Science*, vol. 78, no. 7, pp. 808–817, 2000.

[150] R. Markovinović and J. D. Jansen, “Accelerating iterative solution methods using reduced-order models as solution predictors,” *International Journal for Numerical Methods in Engineering*, vol. 68, no. 5, pp. 525–541, 2006, doi: 10.1002/nme.1721.

[151] M. A. Cardoso, L. J. Durlofsky, and P. Sarma, “Development and application of reduced-order modeling procedures for subsurface flow simulation,” *International Journal for Numerical Methods in Engineering*, vol. 77, no. 9, pp. 1322–1350, 2009, doi: 10.1002/nme.2453.

[152] J. F. M. van Doren, R. Markovinović, and J.-D. Jansen, “Reduced-order optimal control of water flooding using proper orthogonal decomposition,” *Comput Geosci*, vol. 10, no. 1, pp. 137–158, 2006, doi: 10.1007/s10596-005-9014-2.

[153] X. Sun and M. Xu, “Optimal control of water flooding reservoir using proper orthogonal decomposition,” *Journal of Computational and Applied Mathematics*, vol. 320, pp. 120–137, 2017, doi: 10.1016/j.cam.2017.01.020.

[154] M. P. Kaleta, R. G. Hanea, A. W. Heemink, and J.-D. Jansen, “Model-reduced gradient-based history matching,” *Comput Geosci*, vol. 15, no. 1, pp. 135–153, 2011, doi: 10.1007/s10596-010-9203-5.

[155] M. A. A. Cardoso and L. J. J. Durlofsky, “Use of reduced-order modeling procedures for production optimization,” *SPE Journal*, vol. 15, no. 2, pp. 426–435, 2009, doi: 10.2118/119057-PA.

[156] M. A. Cardoso, “Reduced-order models for reservoir simulation,” presented at the SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana: OnePetro, Oct. 2009. doi: 10.2118/129636-STU.

[157] C. Xiao, O. Leeuwenburgh, H. X. Lin, and A. Heemink, “Non-intrusive subdomain POD-TPWL for reservoir history matching,” *Comput Geosci*, vol. 23, no. 3, pp. 537–565, 2019, doi: 10.1007/s10596-018-9803-z.

[158] J. He, P. Sarma, and L. J. Durlofsky, “Reduced-order flow modeling and geological parameterization for ensemble-based data assimilation,” *Computers & Geosciences*, vol. 55, pp. 54–69, 2013, doi: 10.1016/j.cageo.2012.03.027.

[159] M. A. H. Rousset, C. K. Huang, H. Klie, and L. J. Durlofsky, “Reduced-order modeling for thermal recovery processes,” *Comput Geosci*, vol. 18, no. 3, pp. 401–415, 2014, doi: 10.1007/s10596-013-9369-8.

[160] J. He and L. J. Durlofsky, “Reduced-order modeling for compositional simulation using trajectory piecewise linearization,” presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA: OnePetro, Feb. 2013. doi: 10.2118/163634-MS.

[161] E. Gildin, M. Ghasemi, A. Romanovskay, and Y. Efendiev, “Nonlinear complexity reduction for fast simulation of flow in heterogeneous porous media,” presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA: OnePetro, Feb. 2013. doi: 10.2118/163618-MS.

[162] H. Klie, "Unlocking fast reservoir predictions via non-Intrusive reduced order models," presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA: OnePetro, Feb. 2013. doi: 10.2118/163584-MS.

[163] J. Li, X. Fan, Y. Wang, B. Yu, S. Sun, and D. Sun, "A POD-DEIM reduced model for compressible gas reservoir flow based on the Peng-Robinson equation of state," *Journal of Natural Gas Science and Engineering*, vol. 79, 2020, doi: 10.1016/j.jngse.2020.103367.

[164] M. Ghommem, E. Gildin, and M. Ghasemi, "Complexity reduction of multiphase flows in heterogeneous porous media," *SPE Journal*, vol. 21, no. 1, pp. 144–151, 2016, doi: 10.2118/167295-PA.

[165] Z. M. Alghareeb and J. R. Williams, "Optimum decision-making in reservoir managment using reduced-order models," presented at the SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana, USA: OnePetro, Sep. 2013. doi: 10.2118/166305-MS.

[166] S. Yoon, Z. M. Alghareeb, and J. R. Williams, "Hyper-reduced-order models for subsurface flow simulation," *SPE Journal*, vol. 21, no. 6, pp. 2128–2140, 2016, doi: 10.2118/181740-PA.

[167] B. Peherstorfer, D. Butnaru, K. Willcox, and H.-J. Bungartz, "Localized discrete empirical interpolation method," *SIAM J. Sci. Comput.*, vol. 36, no. 1, pp. A168–A192, 2014, doi: 10.1137/130924408.

[168] S. Trehan and L. J. Durlofsky, "Trajectory piecewise quadratic reduced-order model for subsurface flow, with application to PDE-constrained optimization," *Journal of Computational Physics*, vol. 326, pp. 446–473, 2016, doi: 10.1016/j.jcp.2016.08.032.

[169] H. S. Sidhu, A. Narasingam, P. Siddhamshetty, and J. S.-I. Kwon, "Model order reduction of nonlinear parabolic PDE systems with moving boundaries using sparse proper orthogonal decomposition: Application to hydraulic fracturing," *Computers & Chemical Engineering*, vol. 112, pp. 92–100, 2018, doi: 10.1016/j.compchemeng.2018.02.004.

[170] E. Suwartadi, "Gradient-based methods for production optimization of oil reservoirs," PhD dissertation, Norwegian University of Science and Technology, Department of Engineering Cybernetics, 2012.

[171] J. He, "Reduced-order modeling for oil-water and compositional systems, with application to data assimilation and production optimization," PhD dissertation, Stanford University, Department of Energy Resources Engineering, 2013.

[172] A. Golzari, M. Haghighat Sefat, and S. Jamshidi, "Development of an adaptive surrogate model for production optimization," *Journal of Petroleum Science and Engineering*, vol. 133, pp. 677–688, 2015, doi: 10.1016/j.petrol.2015.07.012.

[173] R. Amiri Kolajoobi, H. Haddadpour, and M. Emami Niri, "Investigating the capability of data-driven proxy models as solution for reservoir geological uncertainty quantification," *Journal of Petroleum Science and Engineering*, vol. 205, 2021, doi: 10.1016/j.petrol.2021.108860.

[174] Z. Guo, C. Chen, G. Gao, and J. Vink, "Applying support vector regression to reduce the effect of numerical noise and enhance the performance of history matching," presented at the SPE Annual Technical Conference and Exhibition, San Antonio, Texas, USA: OnePetro, Oct. 2017. doi: 10.2118/187430-MS.

[175] G. D. Avansi, "Use of proxy models in the selection of production strategy and economic evaluation of petroleum fields," presented at the SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana: OnePetro, Oct. 2009. doi: 10.2118/129512-STU.

[176] E. L. Ligero, M. G. Madeira, and D. J. Schiozer, "Comparison of techniques for risk analysis applied to petroleum-field development," presented at the SPE Latin American and Caribbean Petroleum Engineering Conference, Rio de Janeiro, Brazil: OnePetro, Jun. 2005. doi: 10.2118/94806-MS.

[177] F. V. A. Risso, F. F. Risso, and D. J. Schiozer, "Risk assessment of oil fields using proxy models: A case study," *Journal of Canadian Petroleum Technology*, vol. 47, no. 8, 2008, doi: 10.2118/08-08-09-TN.

[178] S. Ghassemzadeh and A. H. Charkhi, "Optimization of integrated production system using advanced proxy based models: A new approach," *Journal of Natural Gas Science and Engineering*, vol. 35, no. Part A, pp. 89–96, 2016, doi: 10.1016/j.jngse.2016.08.045.

[179] G. Zangl, M. Giovannoli, and M. Stundner, "Application of artificial intelligence in gas storage management," presented at the SPE Europec/EAGE Annual Conference and Exhibition, Vienna, Austria: OnePetro, Jun. 2006. doi: 10.2118/100133-MS.

[180] E. Artun, T. Ertekin, R. Watson, and M. Al-Wadhahi, "Development of universal proxy models for screening and optimization of cyclic pressure pulsing in naturally fractured reservoirs," *Journal of Natural Gas Science and Engineering*, vol. 3, no. 6, pp. 667–686, 2011, doi: 10.1016/j.jngse.2011.07.016.

[181] C. Sprunger, T. Muther, F. I. Syed, A. K. Dahaghi, and S. Neghabhan, "State of the art progress in hydraulic fracture modeling using AI/ML techniques," *Model. Earth Syst. Environ.*, vol. 8, no. 1, pp. 1–13, Mar. 2022, doi: 10.1007/s40808-021-01111-w.

[182] F. I. Syed, A. AlShamsi, A. K. Dahaghi, and S. Neghabhan, "Application of ML & AI to model petrophysical and geomechanical properties of shale reservoirs – A systematic literature review," *Petroleum*, vol. 8, no. 2, pp. 158–166, Jun. 2022, doi: 10.1016/j.petlm.2020.12.001.

[183] J. Gu, W. Liu, K. Zhang, L. Zhai, Y. Zhang, and F. Chen, "Reservoir production optimization based on surrograte model and differential evolution algorithm," *Journal of Petroleum Science and Engineering*, vol. 205, 2021, doi: 10.1016/j.petrol.2021.108879.

[184] G. Chen *et al.*, "Surrogate-assisted evolutionary algorithm with dimensionality reduction method for water flooding production optimization," *Journal of Petroleum Science and Engineering*, vol. 185, 2020, doi: 10.1016/j.petrol.2019.106633.

[185] P. Ogbeiwi, K. D. Stephen, and A. O. Arinkoola, "Robust optimisation of water flooding using an experimental design-based surrogate model: A case study of a Niger-Delta oil reservoir," *Journal of Petroleum Science and Engineering*, vol. 195, 2020, doi: 10.1016/j.petrol.2020.107824.

[186] J. Bruyelle and D. Guérillot, "Optimization of waterflooding strategy using artificial neural networks," presented at the SPE Reservoir Characterisation and Simulation Conference and Exhibition, Abu Dhabi, UAE: OnePetro, Sep. 2019. doi: 10.2118/196643-MS.

[187] J. Bruyelle and D. Guérillot, "Well placement optimization with an artificial intelligence method applied to Brugge field," presented at the SPE Gas & Oil Technology Showcase and Conference, Dubai, UAE: OnePetro, Oct. 2019. doi: 10.2118/198656-MS.

[188] H. Hassani, H. Sarkheil, T. Foroud, and S. Karimpooli, "A proxy modeling approach to optimization horizontal well placement," presented at the 45th U.S. Rock Mechanics/Geomechanics Symposium, San Francisco, California: OnePetro, Jun. 2011.

[189] A. Nwachukwu, H. Jeong, M. Pyrcz, and L. W. Lake, "Fast evaluation of well placements in heterogeneous reservoir models using machine learning," *Journal of Petroleum Science and Engineering*, vol. 163, pp. 463–475, 2018, doi: 10.1016/j.petrol.2018.01.019.

[190] H. Aydin, S. Akin, and E. Senturk, "A proxy model for determining reservoir pressure and temperature for geothermal wells," *Geothermics*, vol. 88, 2020, doi: 10.1016/j.geothermics.2020.101916.

[191] L. Wang, Z. Li, C. D. Adenutsi, L. Zhang, F. Lai, and K. Wang, "A novel multi-objective optimization method for well control parameters based on PSO-LSSVR proxy model and NSGA-II algorithm," *Journal of Petroleum Science and Engineering*, vol. 196, 2021, doi: 10.1016/j.petrol.2020.107694.

[192] E. Fedutenko, C. Yang, C. Card, and L. X. Nghiem, "Time-dependent proxy modeling of SAGD process," presented at the SPE Heavy Oil Conference-Canada, Calgary, Alberta, Canada: OnePetro, Jun. 2013. doi: 10.2118/165395-MS.

[193] W. J. Al-Mudhafar and D. N. Rao, "Proxy-based metamodeling optimization of the gas-assisted gravity drainage GAGD process in heterogeneous sandstone reservoirs," presented at the SPE Western Regional Meeting, Bakersfield, California: OnePetro, Apr. 2017. doi: 10.2118/185701-MS.

[194] A. K. Jaber, M. B. Awang, and C. P. Lenn, "Box-Behnken design for assessment proxy model of miscible CO2-WAG in heterogeneous clastic reservoir," *Journal of Natural Gas Science and Engineering*, vol. 40, no. C, pp. 236–248, 2017, doi: 10.1016/j.jngse.2017.02.020.

[195] S. Agada, S. Geiger, A. Elsheikh, and S. Oladyshkin, "Data-driven surrogates for rapid simulation and optimization of WAG injection in fractured carbonate reservoirs," *Petroleum Geoscience*, vol. 23, no. 2, pp. 270–283, 2017, doi: 10.1144/petgeo2016-068.

[196] E. Ibiam, S. Geiger, V. Demyanov, and D. Arnold, "Optimization of polymer flooding in a heterogeneous reservoir considering geological and history matching uncertainties," *SPE Reservoir Evaluation & Engineering*, vol. 24, no. 01, pp. 19–36, 2021, doi: 10.2118/200568-PA.

[197] F. Alenezi and S. Mohaghegh, "A data-driven smart proxy model for a comprehensive reservoir simulation," presented at the 4th Saudi International Conference on Information Technology (Big Data Analysis) (KACSTIT), Nov. 2016, pp. 1–6. doi: 10.1109/KACSTIT.2016.7756063.

[198] X. Tang, L. A. James, and T. E. Johansen, "A new streamline model for near-well flow validated with radial flow experiments," *Comput Geosci*, vol. 22, no. 1, pp. 363–388, Feb. 2018, doi: 10.1007/s10596-017-9697-1.

[199] S. A. Smolyak, "Quadrature and interpolation formulas for tensor products of certain classes of functions," *Soviet Mathematics Doklady*, vol. 4, pp. 240–243, 1963.

[200] G. Gao, J. C. Vink, C. Chen, M. Tarrahi, and Y. El Khamra, "Uncertainty quantification for history matching problems with multiple best matches using a distributed Gauss-Newton method," presented at the SPE Annual Technical Conference and Exhibition, Dubai, UAE: OnePetro, Sep. 2016. doi: 10.2118/181611-MS.

[201] N. Hansen and A. Ostermeier, "Completely derandomized self-Adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001, doi: 10.1162/106365601750190398.

[202] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002, doi: 10.1109/4235.996017.

[203] M. Simonov, A. Shubin, A. Penigin, D. Perets, E. Belonogov, and A. Margarit, “Optimization of oil field development using a surrogate model: Case of miscible gas injection,” presented at the SPE Reservoir Characterisation and Simulation Conference and Exhibition, Abu Dhabi, UAE: OnePetro, Sep. 2019. doi: 10.2118/196639-MS.

[204] K. Redouane, N. Zeraibi, and M. Nait Amar, “Automated optimization of well placement via adaptive space-filling surrogate modelling and evolutionary algorithm,” presented at the Abu Dhabi International Petroleum Exhibition & Conference, Abu Dhabi, UAE: OnePetro, Nov. 2018. doi: 10.2118/193040-MS.

[205] J.-S. R. Jang, “ANFIS: adaptive-network-based fuzzy inference system,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993, doi: 10.1109/21.256541.

[206] A. H. Elsheikh, I. Hoteit, and M. F. Wheeler, “Efficient Bayesian inference of subsurface flow models using nested sampling and sparse polynomial chaos surrogates,” *Computer Methods in Applied Mechanics and Engineering*, vol. 269, pp. 515–537, 2014, doi: 10.1016/j.cma.2013.11.001.

[207] T. Yu, D. Wilkinson, and A. Castellini, “Constructing reservoir flow simulator proxies using genetic programming for history matching and production forecast uncertainty analysis,” *Journal of Artificial Evolution and Applications*, vol. 2008, p. e263108, 2007, doi: 10.1155/2008/263108.

[208] J. M. Hammersley, “Monte Carlo methods for solving multivariable problems,” *Annals of the New York Academy of Sciences*, vol. 86, no. 3, pp. 844–874, 1960, doi: 10.1111/j.1749-6632.1960.tb42846.x.

[209] S. Kalla and C. D. White, “Efficient design of reservoir simulation studies for development and optimization,” presented at the SPE Annual Technical Conference and Exhibition, Dallas, Texas: OnePetro, Oct. 2005. doi: 10.2118/95456-MS.

[210] Y. D. Kim and L. J. Durlofsky, “Convolutional-recurrent neural network proxy for robust optimization and closed-Loop reservoir management.” arXiv, Jun. 01, 2022. doi: 10.48550/arXiv.2203.07524.

[211] Y. D. Kim and L. J. Durlofsky, “A recurrent neural network–based proxy model for well-control optimization with nonlinear output constraints,” *SPE Journal*, vol. 26, no. 04, pp. 1837–1857, Aug. 2021, doi: 10.2118/203980-PA.

[212] J. Kim, H. Yang, and J. Choe, “Robust optimization of the locations and types of multiple wells using CNN based proxy models,” *Journal of Petroleum Science and Engineering*, vol. 193, p. 107424, Oct. 2020, doi: 10.1016/j.petrol.2020.107424.

[213] T. Austad, “Chapter 13 - water-based EOR in carbonates and sandstones: new chemical understanding of the EOR potential using ‘smart water,’” in *Enhanced Oil Recovery Field Case Studies*, J. J. Sheng, Ed., Boston: Gulf Professional Publishing, 2013, pp. 301–335. doi: 10.1016/B978-0-12-386545-8.00013-0.

[214] A. O. Arinkoola, U. I. Duru, and H. M. Onuh, “Development of proxy model for production forecast using adaptive neuro-fuzzy inference system and experimental design,” *IJPE*, vol. 1, no. 3, p. 189, 2015, doi: 10.1504/IJPE.2015.071062.

[215] P. Bahrami, F. Sahari Moghaddam, and L. A. James, “A review of proxy modeling highlighting applications for reservoir engineering,” *Energies*, vol. 15, no. 14, Art. no. 14, Jan. 2022, doi: 10.3390/en15145247.

[216] G. Assaf, M. Heiner, and F. Liu, “Fuzzy Petri nets and coloured fuzzy Petri nets in Snoopy,” Computer Science Department Brandenburg University of Technology Cottbus-Senftenberg, Jan. 2021.

[217] J. Brownlee, *Data preparation for machine learning - data cleaning, feature selection, and data transforms in Python*. Machine Learning Mastery, 2020.

[218] R. Arun Kumar, J. Vijay Franklin, and N. Koppula, “A comprehensive survey on metaheuristic algorithm for feature selection techniques,” *Materials Today: Proceedings*, May 2022, doi: 10.1016/j.matpr.2022.04.803.

[219] B. Xue, M. Zhang, W. N. Browne, and X. Yao, “A survey on evolutionary computation approaches to feature selection,” *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 4, pp. 606–626, Aug. 2016, doi: 10.1109/TEVC.2015.2504420.

[220] J. Szlęk, A. Pacławski, R. Lau, R. Jachowicz, P. Kazemi, and A. Mendyk, “Empirical search for factors affecting mean particle size of PLGA microspheres containing macromolecular drugs,” *Computer Methods and Programs in Biomedicine*, vol. 134, pp. 137–147, 2016, doi: 10.1016/j.cmpb.2016.07.006.

[221] J. Szlęk, A. Pacławski, R. Lau, R. Jachowicz, and A. Mendyk, “Heuristic modeling of macromolecule release from PLGA microspheres,” *IJN*, vol. 8, no. 1, pp. 4601–4611, Dec. 2013, doi: 10.2147/IJN.S53364.

[222] A. Bhandare, M. Bhide, P. Gokhale, and R. Chandavarkar, “Applications of Convolutional Neural Networks,” *International Journal of Computer Science and Information Technologies*, vol. 7, no. 5, pp. 2206–2215, 2016.

[223] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006, doi: 10.1162/neco.2006.18.7.1527.

[224] M. Bosch, “Intelligent static and dynamic reservoir model building based on knowledge networks,” presented at the SEG/AAPG International Meeting for Applied Geoscience & Energy, OnePetro, Nov. 2022. doi: 10.1190/image2022-3751503.1.

[225] P. Q. Memon, S.-P. Yong, W. Pao, and P. J. Sean, “A preliminary study on well-based surrogate reservoir model,” in *2013 IEEE Student Conference on Research and Developement*, Dec. 2013, pp. 305–310. doi: 10.1109/SCOReD.2013.7002595.

[226] C. A. Visser and J. L. S. Viota, “Introduction to the Groningen static reservoir model,” *Netherlands Journal of Geosciences*, vol. 96, no. 5, pp. s39–s46, Dec. 2017, doi: 10.1017/njg.2017.25.

[227] L. B. Cunha, “Integrating static and dynamic data for oil and gas reservoir modelling,” *Journal of Canadian Petroleum Technology*, vol. 43, no. 03, Mar. 2004, doi: 10.2118/04-03-TN.

[228] A. Shahkarami, S. Mohaghegh, V. Gholami, A. Haghighat, and D. Moreno, “Modeling pressure and saturation distribution in a CO2 storage project using a Surrogate Reservoir Model (SRM),” *Greenhouse Gases: Science and Technology*, vol. 4, no. 3, pp. 289–315, Jun. 2014, doi: 10.1002/ghg.1414.

[229] R. Stocki, “A method to improve design reliability using optimal Latin hypercube sampling,” *Computer Assisted Mechanics and Engineering Sciences*, no. Vol. 12, No. 4, pp. 393–411, 2005.

[230] Y. Saeys, T. Abeel, and Y. Van de Peer, “Robust Feature Selection Using Ensemble Feature Selection Techniques,” in *Machine Learning and Knowledge Discovery in Databases*, W. Daelemans, B. Goethals, and K. Morik, Eds., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2008, pp. 313–325. doi: 10.1007/978-3-540-87481-2_21.

[231] P. Bahrami and L. A. James, “Screening of waterflooding using smart proxy model coupled with deep convolutional neural network,” *Geoenergy Science and Engineering*, vol. 221, p. 111300, Feb. 2023, doi: 10.1016/j.petrol.2022.111300.

[232] N. Abdullah and N. Hasan, “The implementation of Water Alternating (WAG) injection to obtain optimum recovery in Cornea Field, Australia,” *J Petrol Explor Prod Technol*, vol. 11, no. 3, pp. 1475–1485, Mar. 2021, doi: 10.1007/s13202-021-01103-7.

[233] M. A. Samba and M. O. Elsharafi, “Literature Review of Water Alternation Gas Injection,” *JEEE*, vol. 7, no. 2, Art. no. 2, Oct. 2018, doi: 10.25299/jeee.2018.vol7(2).2117.

[234] D. N. Rao and S. G. Sayegh, “Timing of miscible hydrocarbon gas injection after waterflooding,” *J Can Pet Technol*, vol. 31, no. 6, 1992, doi: https://doi.org/10.2118/92-06-05.

[235] B. Chen and A. C. Reynolds, “CO2 water-alternating-gas injection for enhanced oil recovery: Optimal well controls and half-cycle lengths,” *Computers & Chemical Engineering*, vol. 113, pp. 44–56, May 2018, doi: 10.1016/j.compchemeng.2018.03.006.

[236] G. R. Jerauld, “Timing of miscible hydrocarbon gas injection after waterflooding,” presented at the SPE/DOE Improved Oil Recovery Symposium, Tulsa, Oklahoma, Apr. 2000. doi: Tulsa, Oklahoma.

Appendix A

The results of grid-based SPM for the 20th, 40th, and 63rd layers at 1st, 10th, and 20th timesteps are presented in this appendix. These results were obtained for the Volve field when it was under waterflooding.

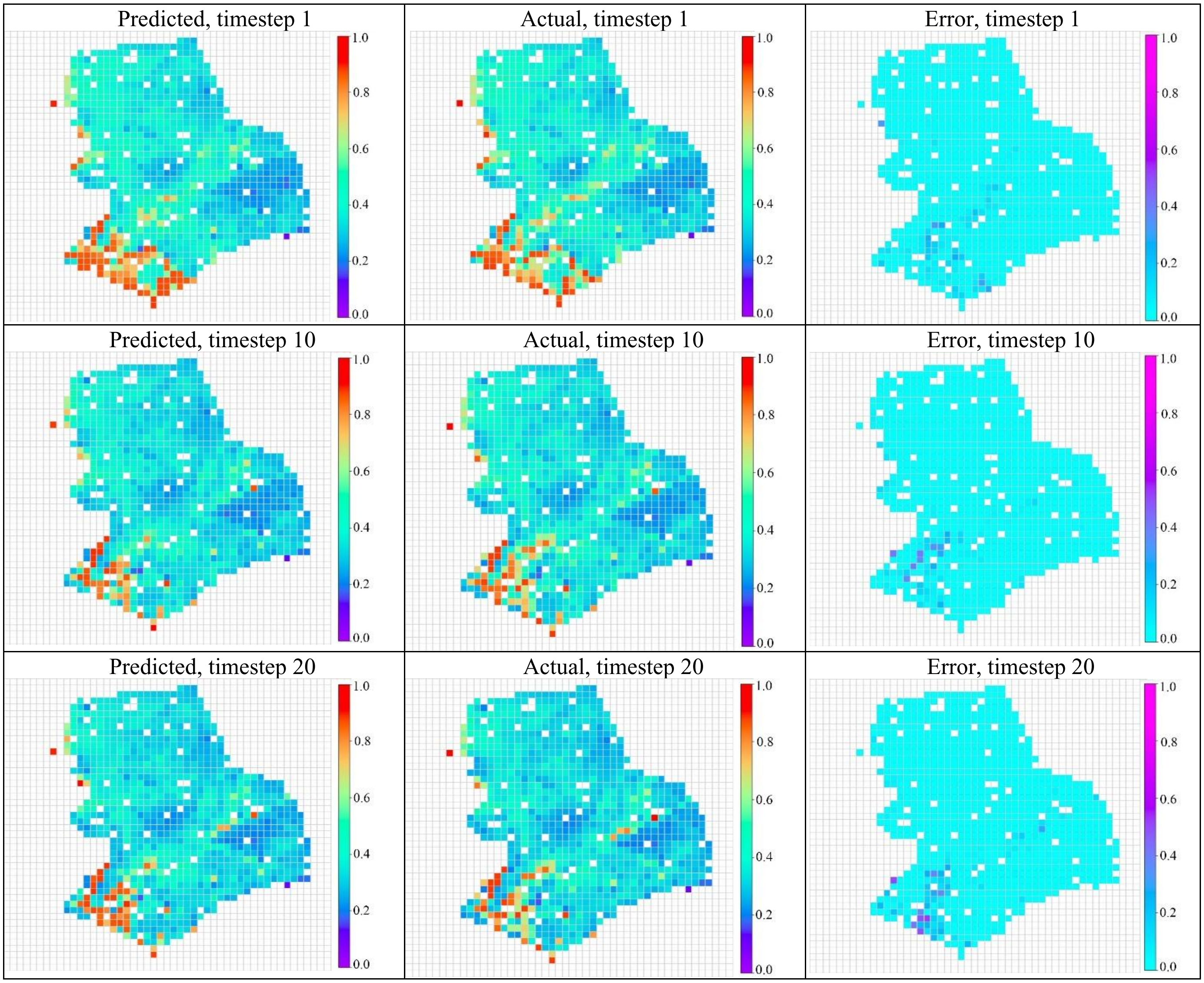


Figure 54. Grids oil saturation in layer 20 for the blind test (waterflooding case)

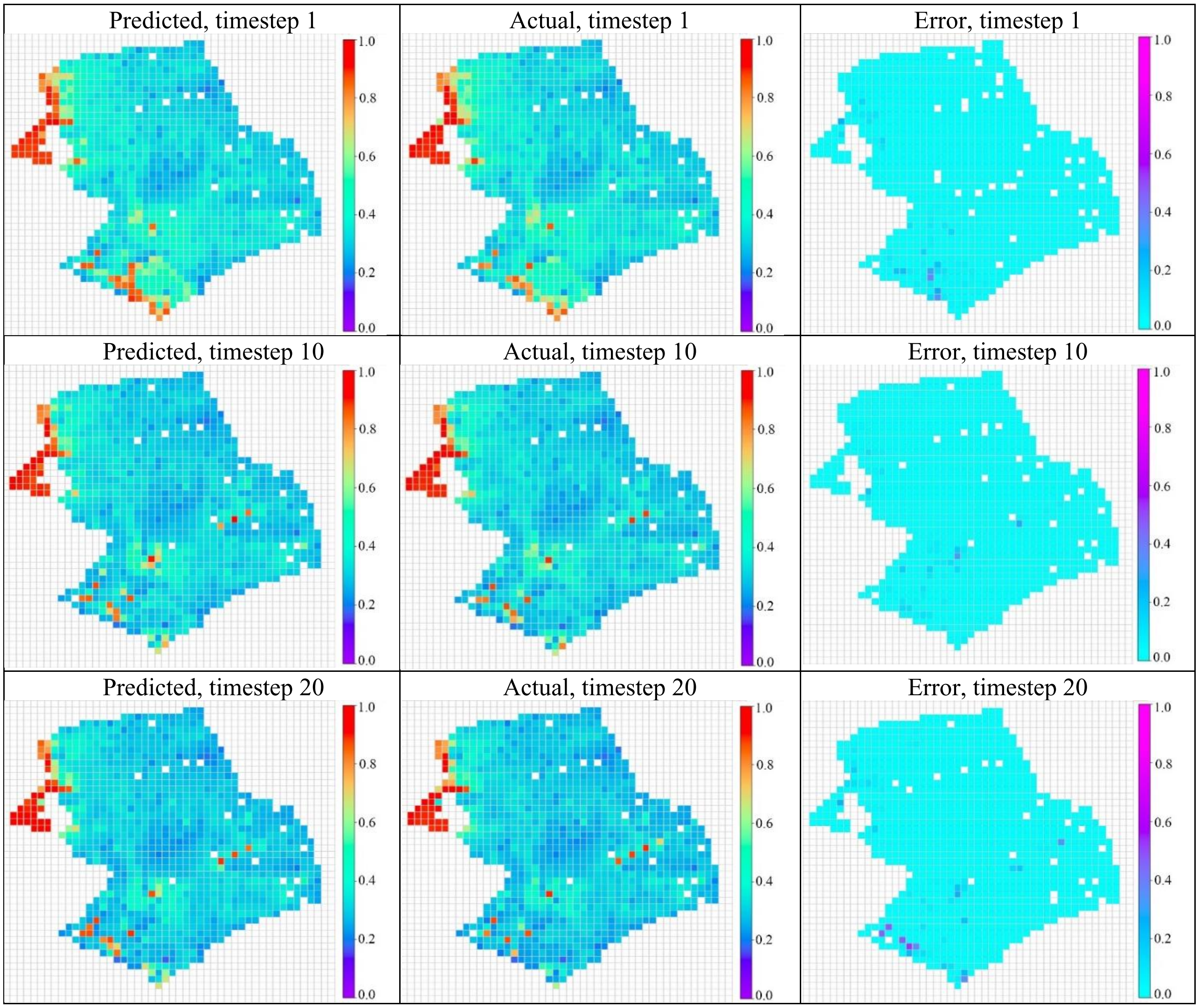


Figure 55. Grids oil saturation in layer 40 for the blind test (waterflooding case)

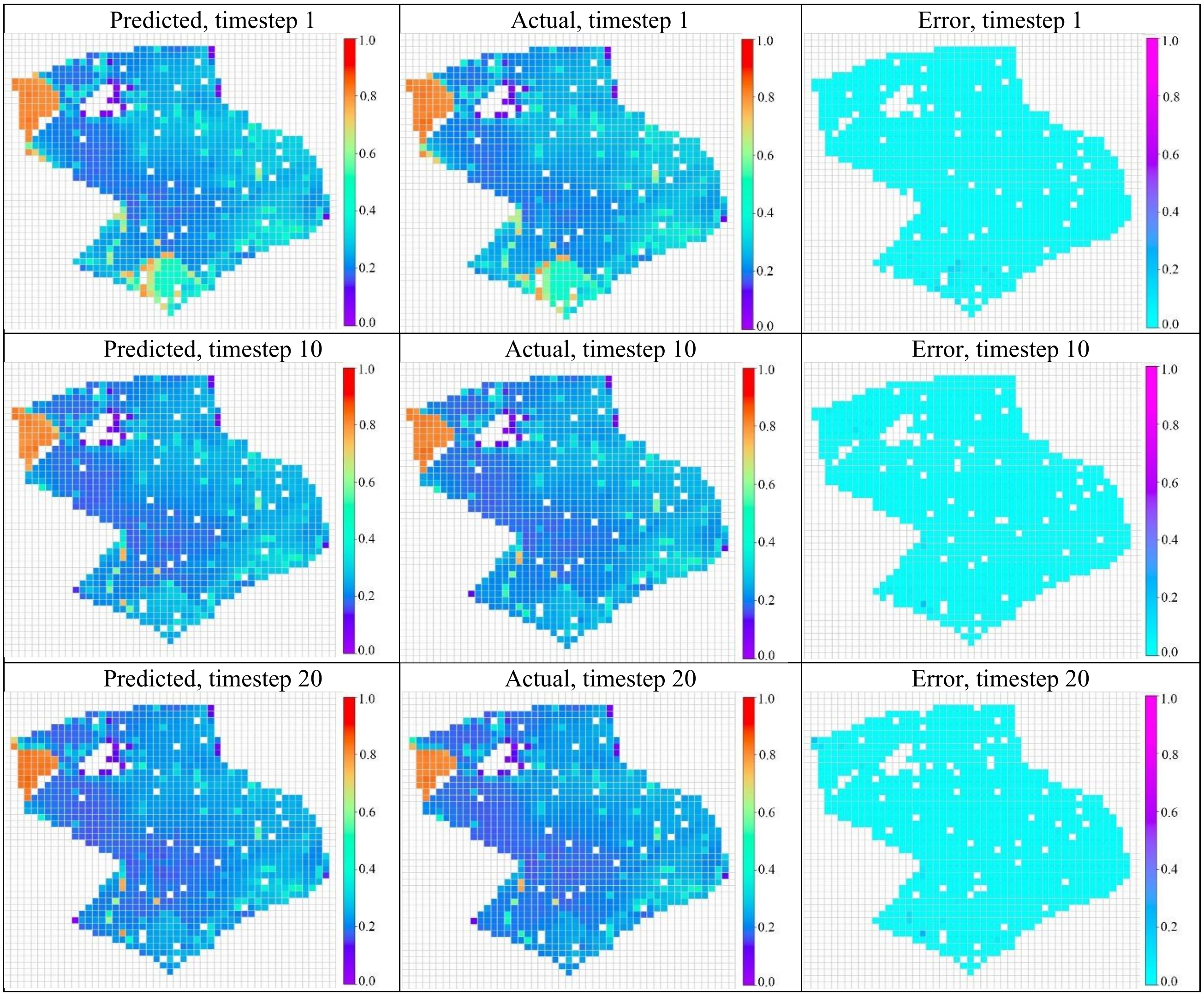


Figure 56. Grids oil saturation in layer 63 for the blind test (waterflooding case)

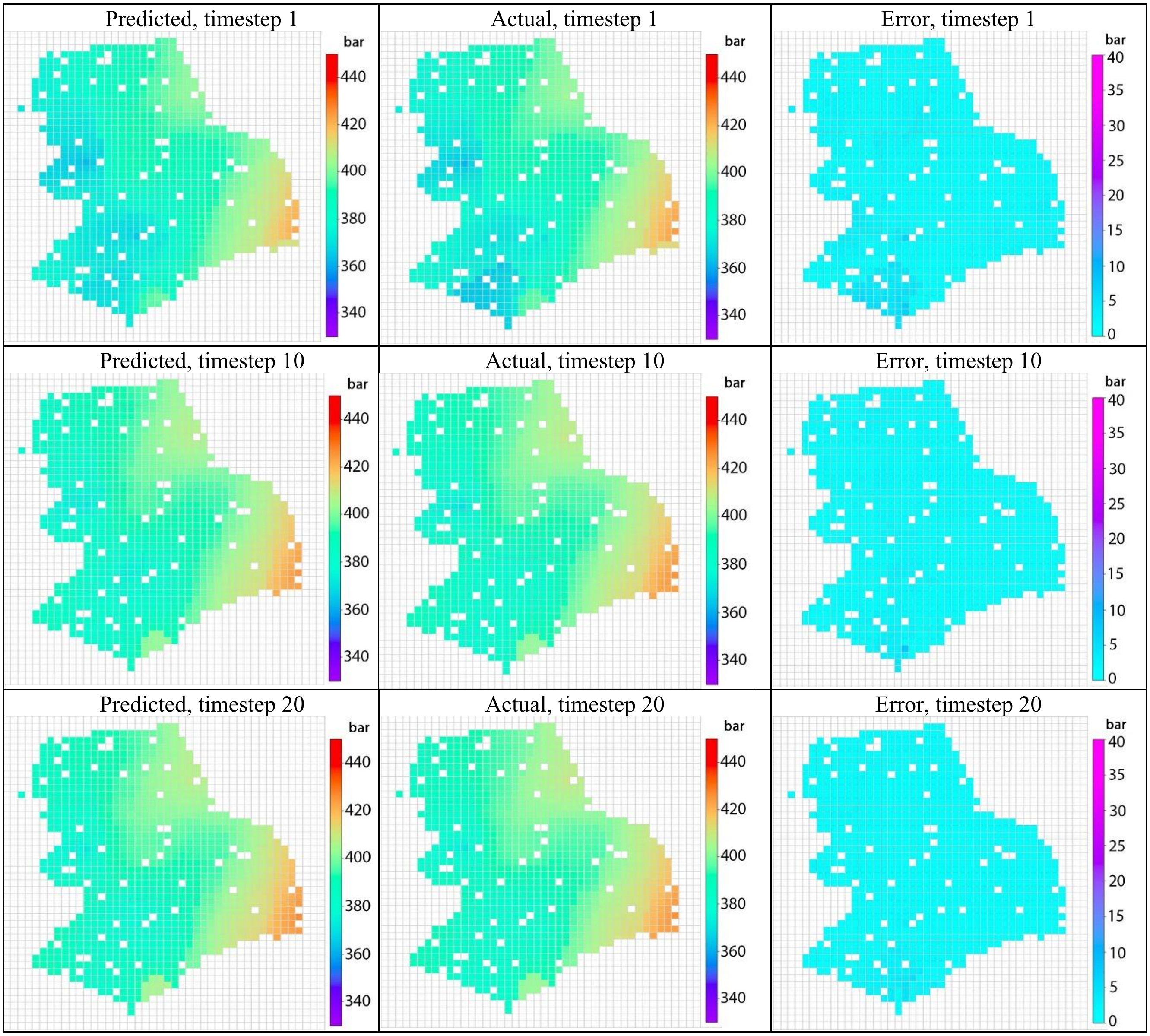


Figure 57. Grids pressure in layer 20 for the blind test (waterflooding case)

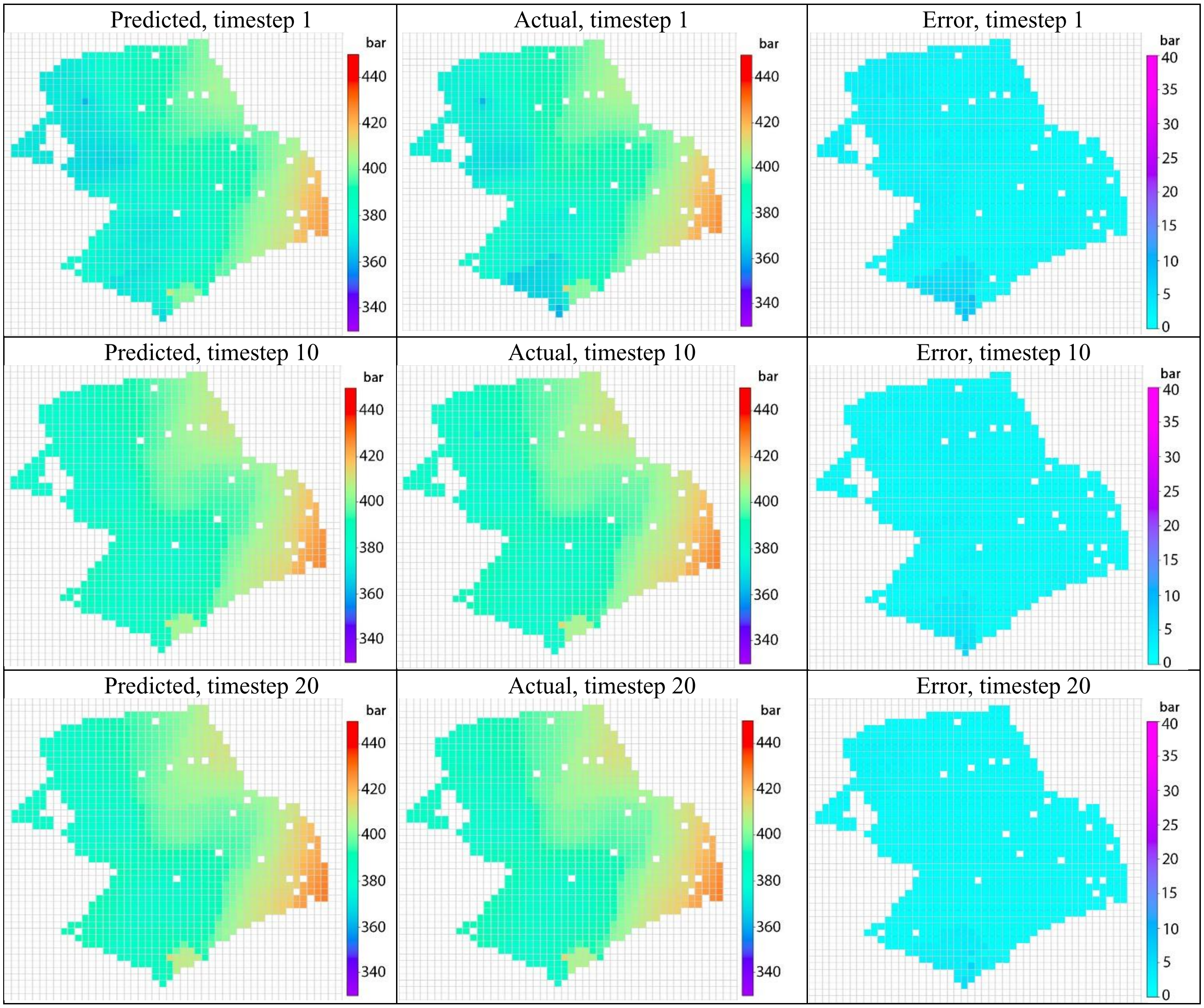


Figure 58. Grids pressure in layer 40 for the blind test (waterflooding case)

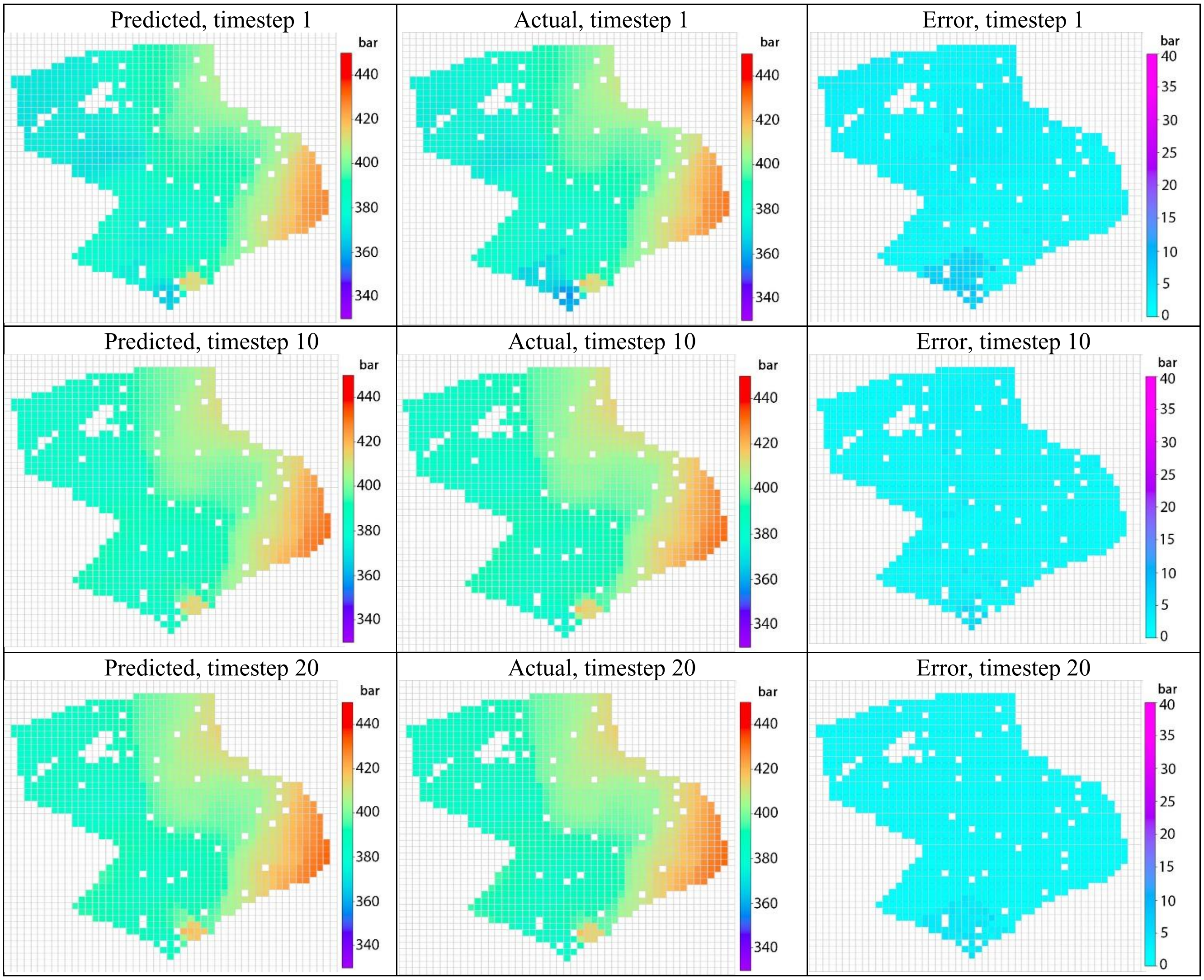
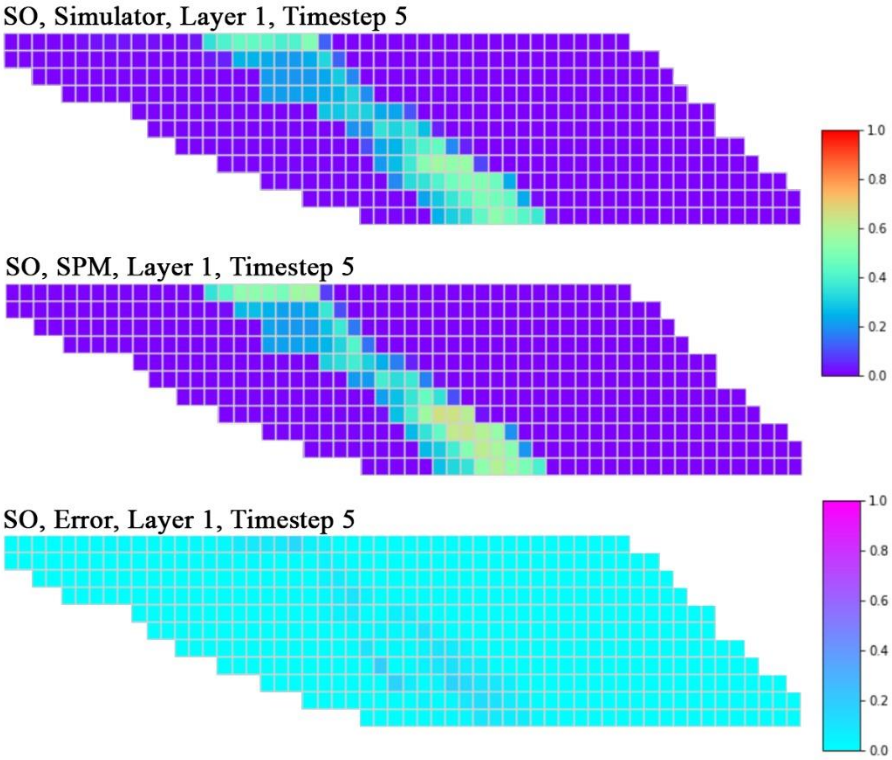


Figure 59. Grids pressure in layer 63 for the blind test (waterflooding case)

Appendix B

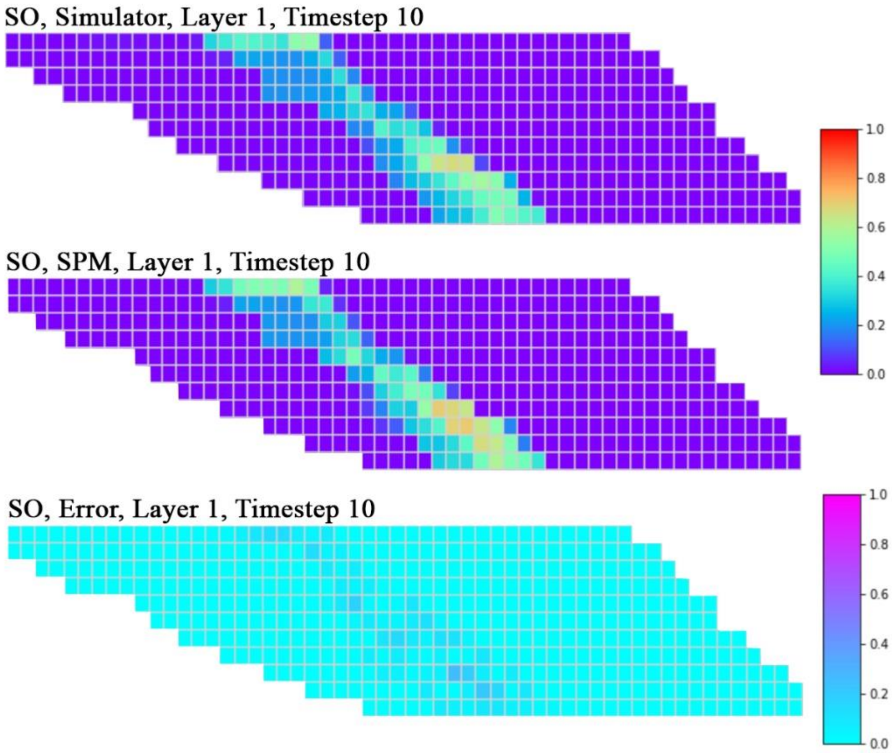
The results of grid-based SPM for the 1st, 5th, 10th, and 15th layers at 5th, 10th, 15th, and 21st timesteps are presented in this appendix. These results were obtained for the Norne E-section field when it was under WAG injection.


SO, Simulator, Layer 1, Timestep 5

SO, SPM, Layer 1, Timestep 5

SO, Error, Layer 1, Timestep 5

Figure 60. Grids oil saturation in L1 and T5 (Sept 2006) for the blind test (WAG case)


SO, Simulator, Layer 1, Timestep 10

SO, SPM, Layer 1, Timestep 10

SO, Error, Layer 1, Timestep 10

Figure 61. Grids oil saturation in L1 and T10 (May 2008) for the blind test (WAG case)

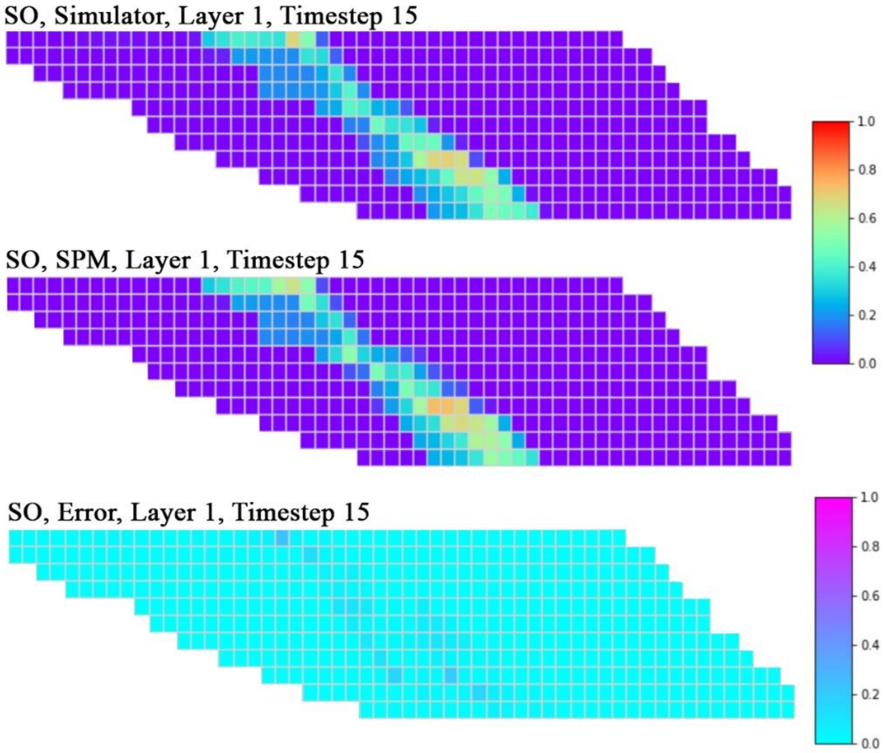


Figure 62. Grids oil saturation in L1 and T15 (Jan 2010) for the blind test (WAG case)

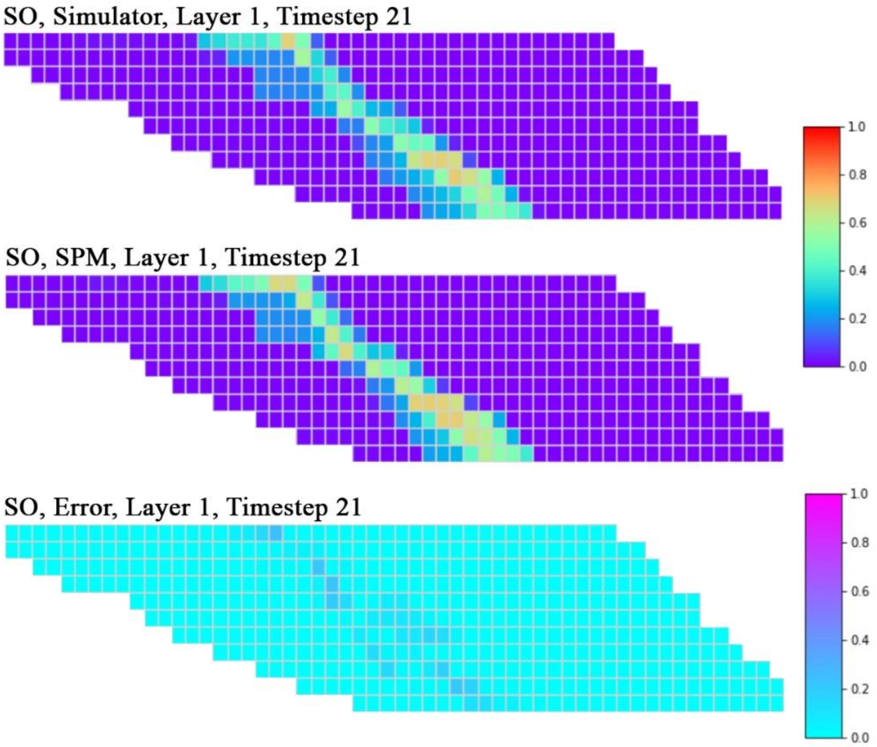
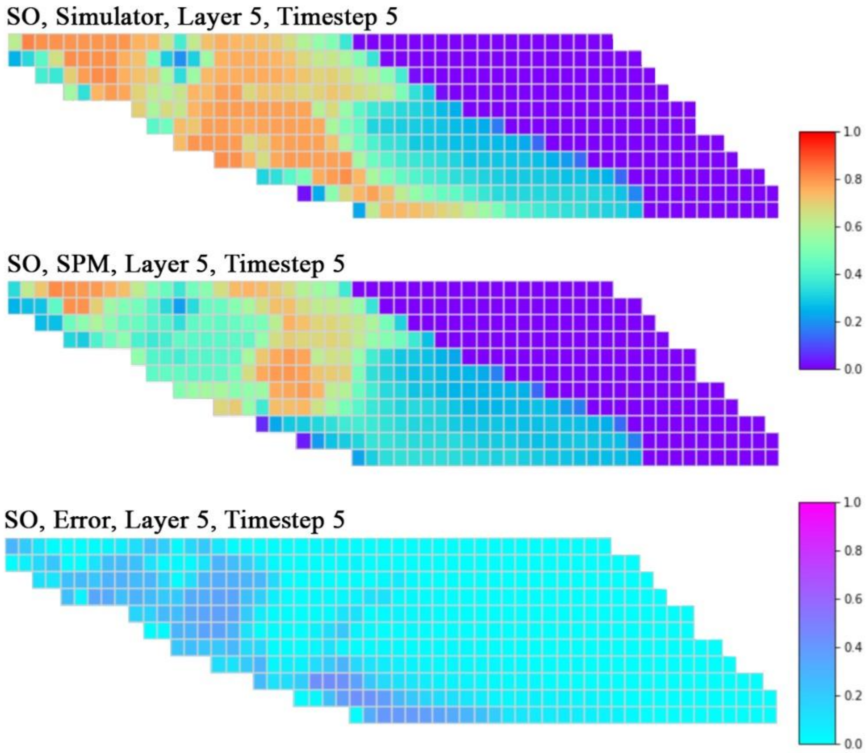


Figure 63. Grids oil saturation in L1 and T21 (Jan 2012) for the blind test (WAG case)

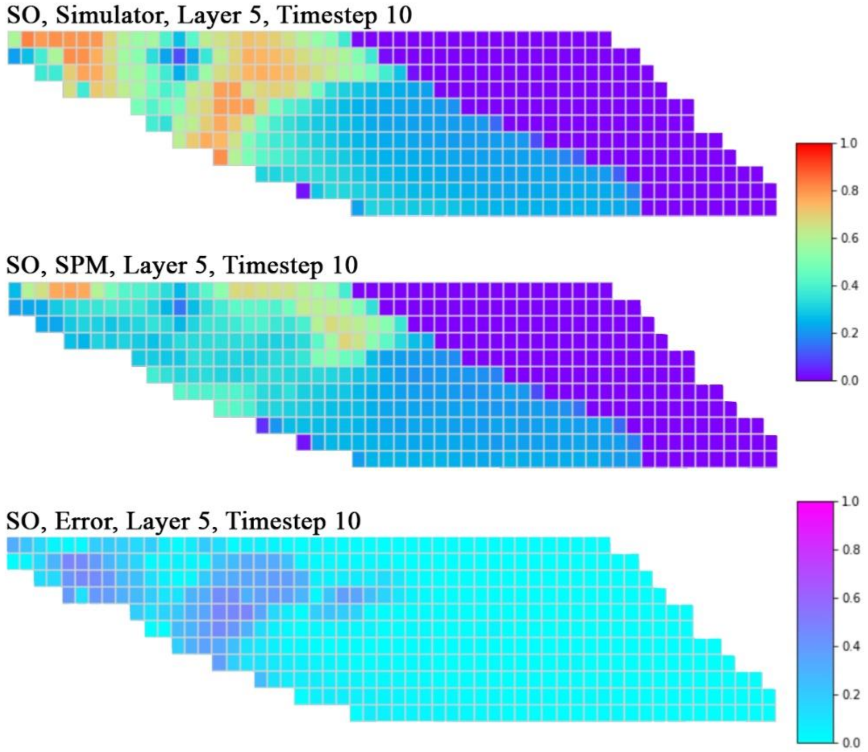


SO, Simulator, Layer 5, Timestep 5

SO, SPM, Layer 5, Timestep 5

SO, Error, Layer 5, Timestep 5

Figure 64. Grids oil saturation in L5 and T5 (Sept 2006) for the blind test (WAG case)



SO, Simulator, Layer 5, Timestep 10

SO, SPM, Layer 5, Timestep 10

SO, Error, Layer 5, Timestep 10

Figure 65. Grids oil saturation in L5 and T10 (May 2008) for the blind test (WAG)

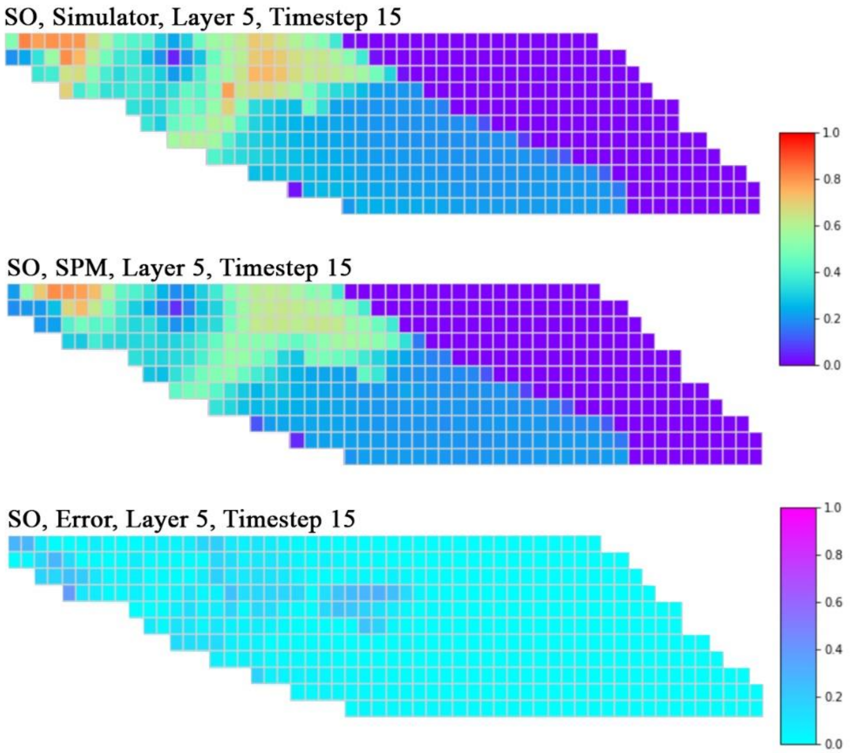


Figure 66. Grids oil saturation in L5 and T15 (Jan 2010) for the blind test (WAG case)

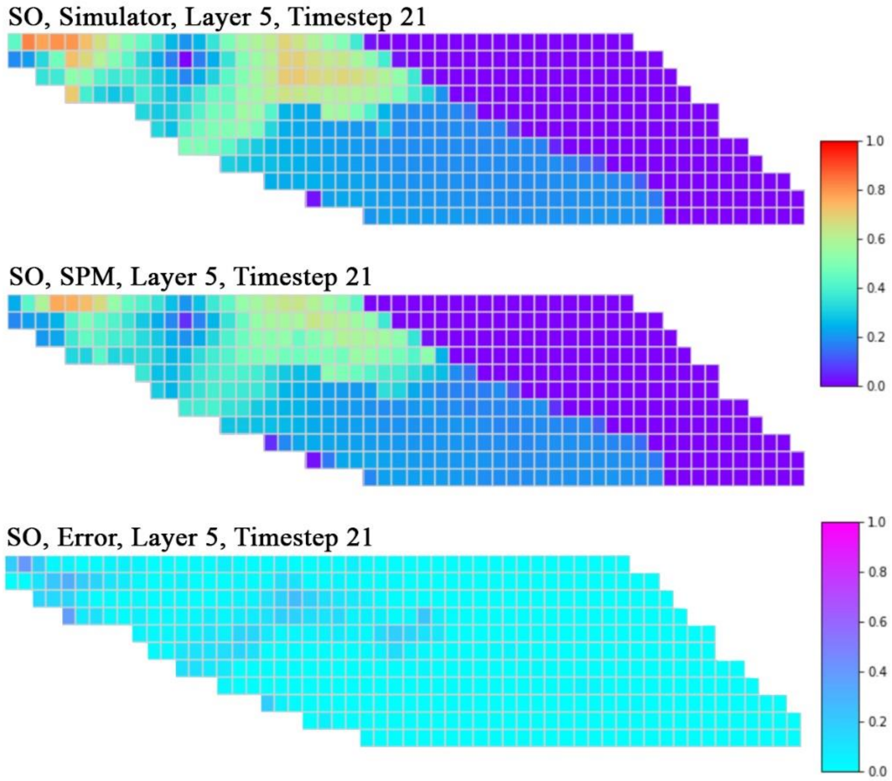
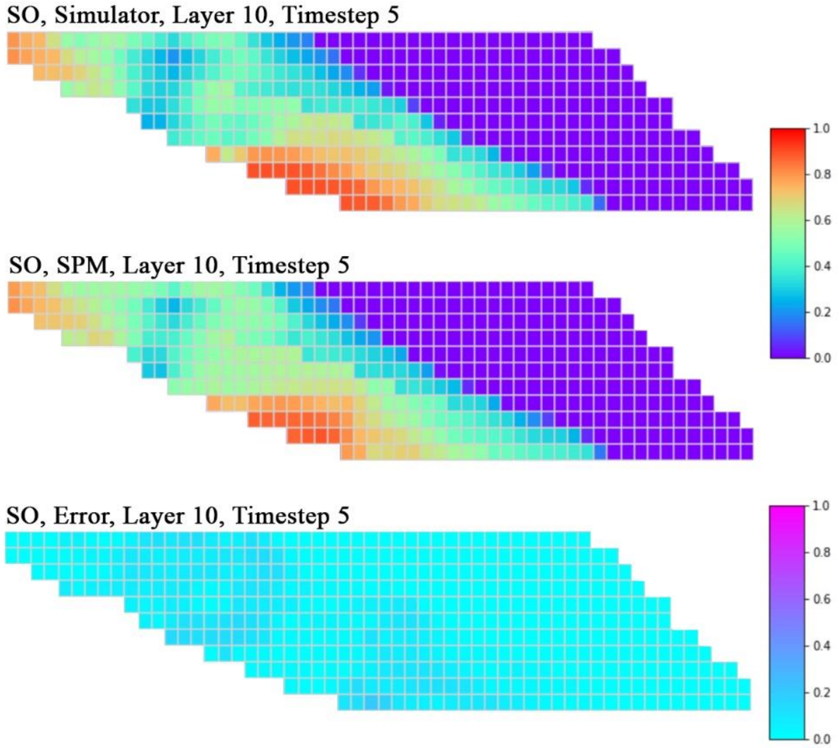


Figure 67. Grids oil saturation in L5 and T21 (Jan 2012) for the blind test (WAG case)

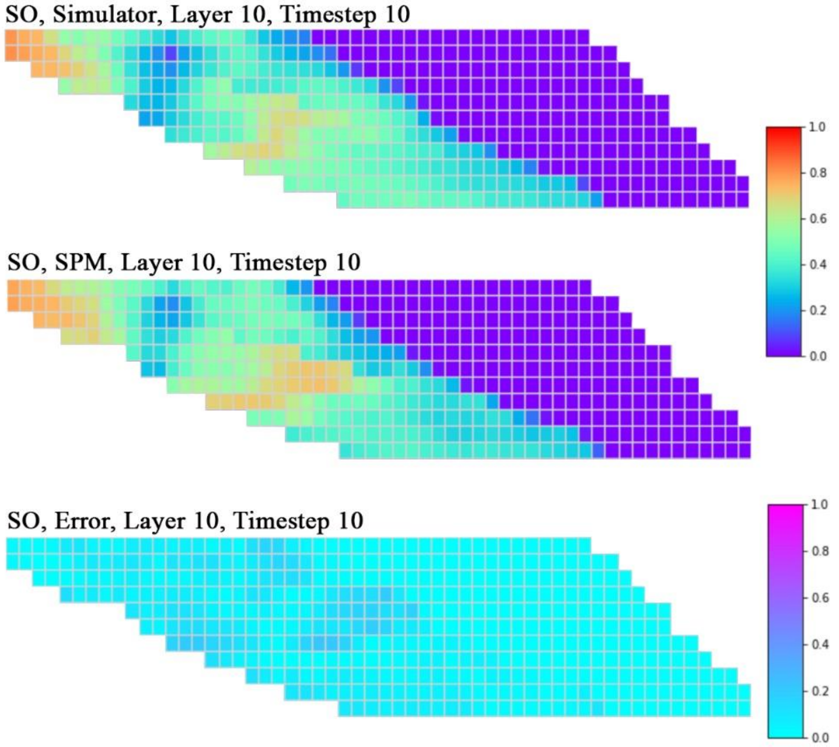


SO, Simulator, Layer 10, Timestep 5

SO, SPM, Layer 10, Timestep 5

SO, Error, Layer 10, Timestep 5

Figure 68. Grids oil saturation in L10 and T5 (Sept 2006) for the blind test (WAG case)

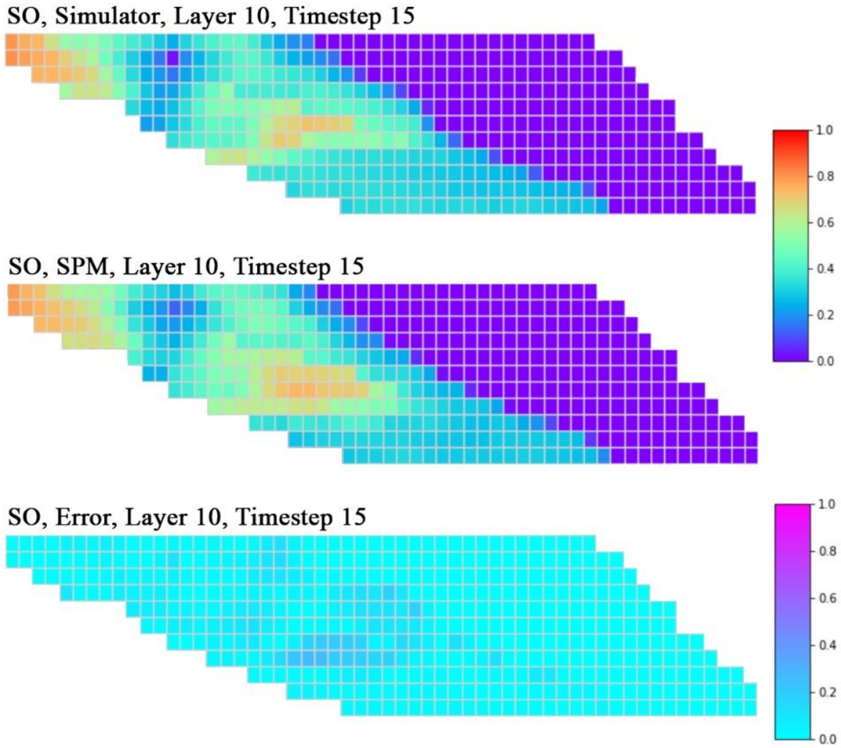


SO, Simulator, Layer 10, Timestep 10

SO, SPM, Layer 10, Timestep 10

SO, Error, Layer 10, Timestep 10

Figure 69. Grids oil saturation in L10 and T10 (May 2008) for the blind test (WAG case)

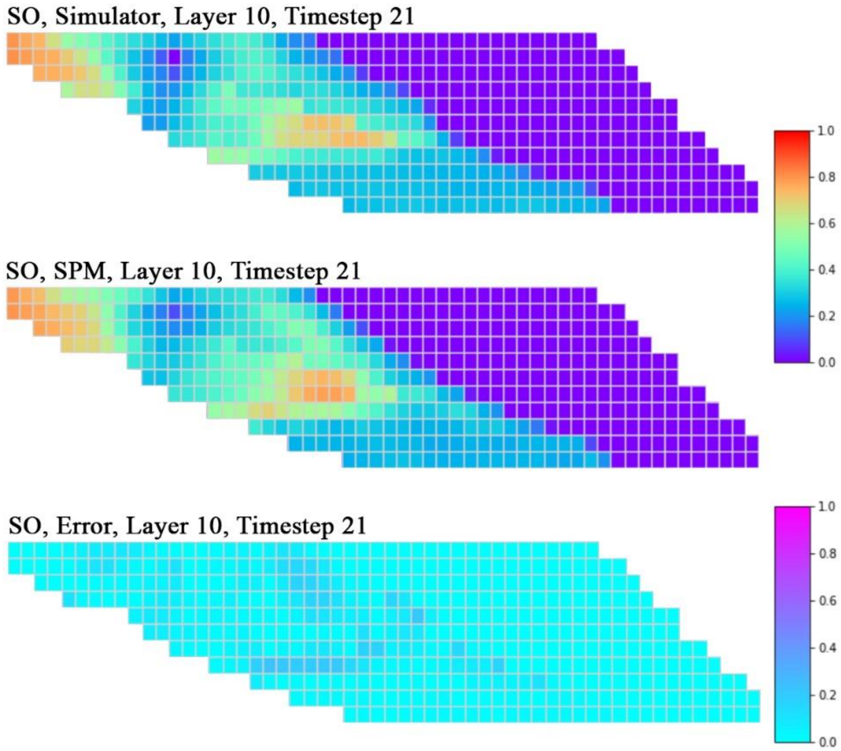


SO, Simulator, Layer 10, Timestep 15

SO, SPM, Layer 10, Timestep 15

SO, Error, Layer 10, Timestep 15

Figure 70. Grids oil saturation in L10 and T15 (Jan 2010) for the blind test (WAG case)

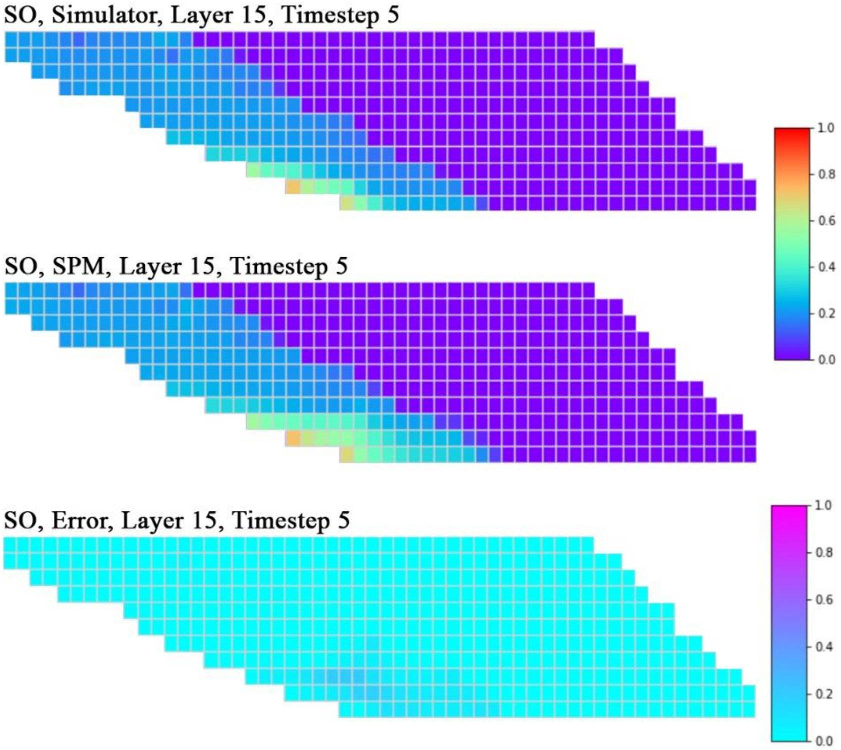


SO, Simulator, Layer 10, Timestep 21

SO, SPM, Layer 10, Timestep 21

SO, Error, Layer 10, Timestep 21

Figure 71. Grids oil saturation in L10 and T21 (Jan 2012) for the blind test (WAG case)

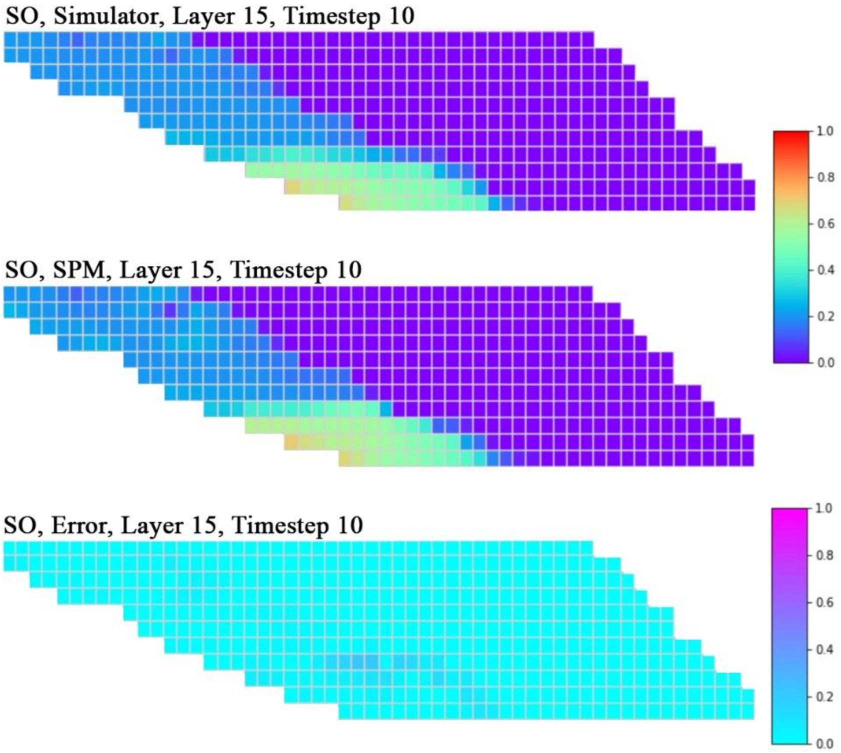


SO, Simulator, Layer 15, Timestep 5

SO, SPM, Layer 15, Timestep 5

SO, Error, Layer 15, Timestep 5

Figure 72. Grids oil saturation in L15 and T5 (Sept 2006) for the blind test (WAG case)

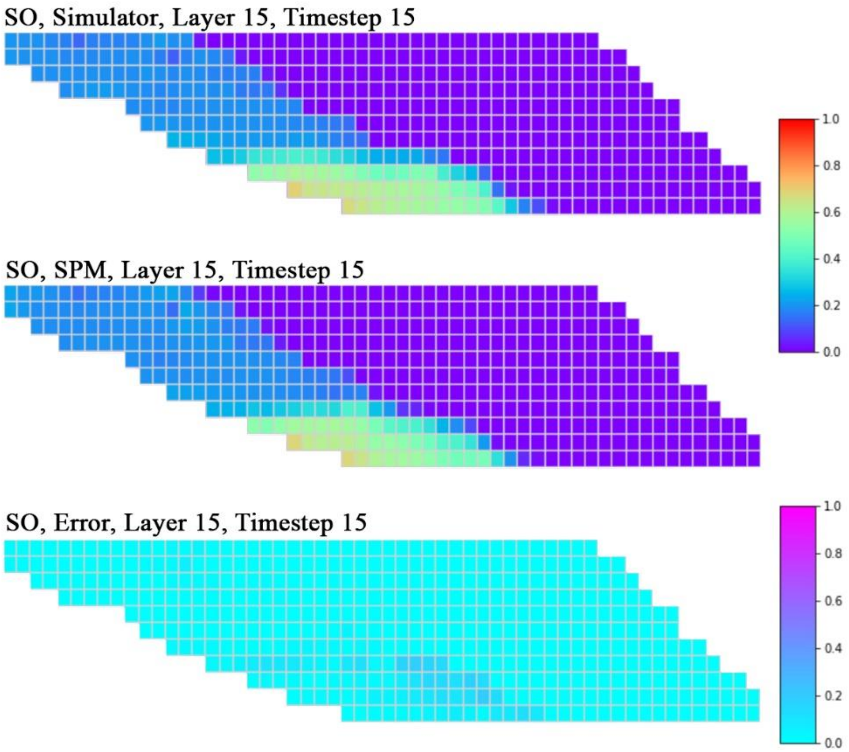


SO, Simulator, Layer 15, Timestep 10

SO, SPM, Layer 15, Timestep 10

SO, Error, Layer 15, Timestep 10

Figure 73. Grids oil saturation in L15 and T10 (May 2008) for the blind test (WAG case)

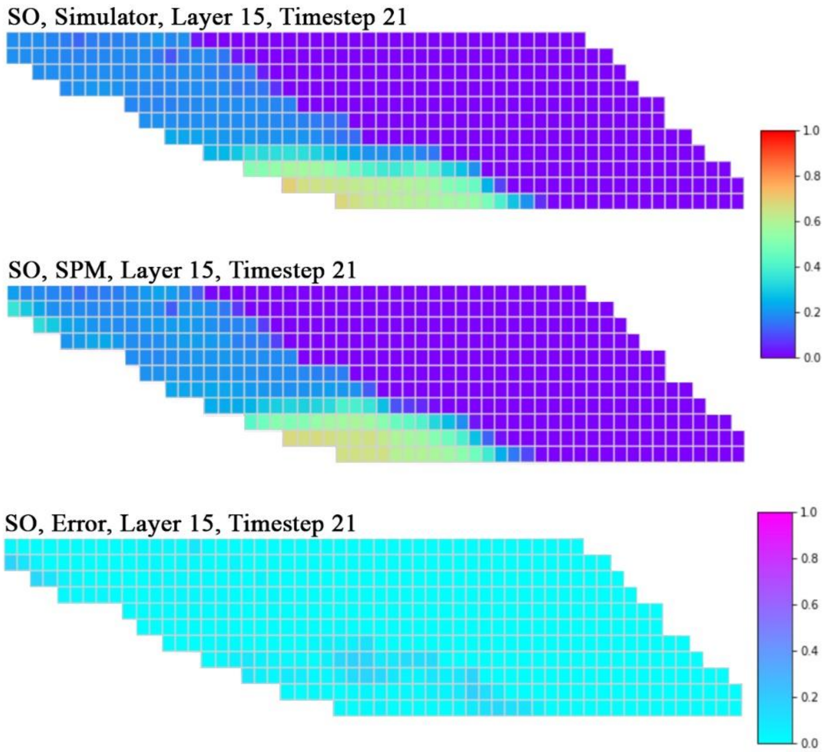


SO, Simulator, Layer 15, Timestep 15

SO, SPM, Layer 15, Timestep 15

SO, Error, Layer 15, Timestep 15

Figure 74. Grids oil saturation in L15 and T15 (Jan 2010) for the blind test (WAG case)



SO, Simulator, Layer 15, Timestep 21

SO, SPM, Layer 15, Timestep 21

SO, Error, Layer 15, Timestep 21

Figure 75. Grids oil saturation in L15 and T21 (Jan 2012) for the blind test (WAG case)

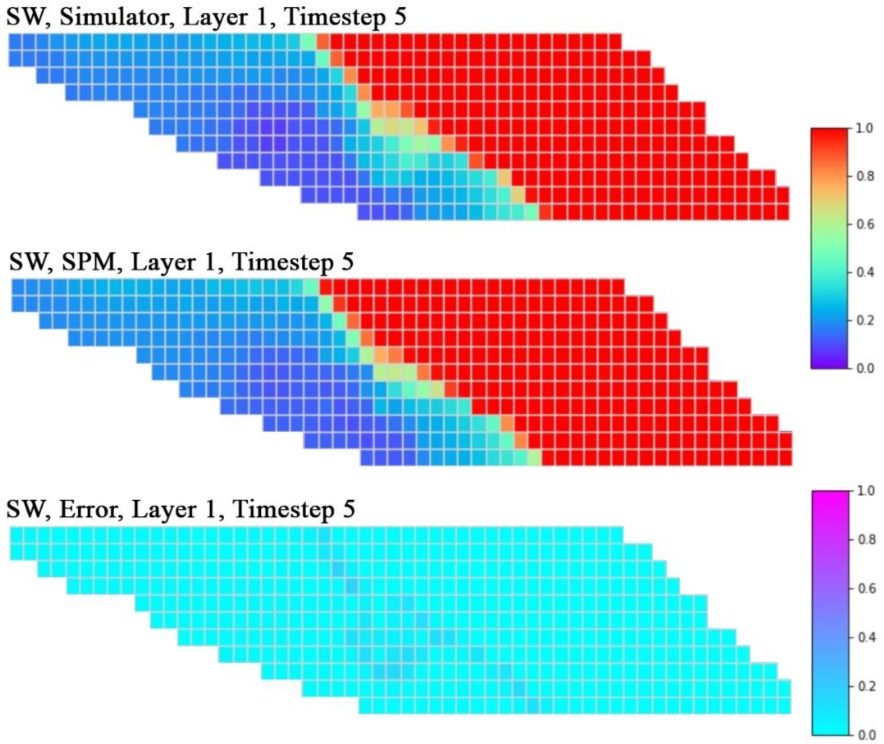


Figure 76. Grids water saturation in L1 and T5 (Sept 2006) for the blind test (WAG case)

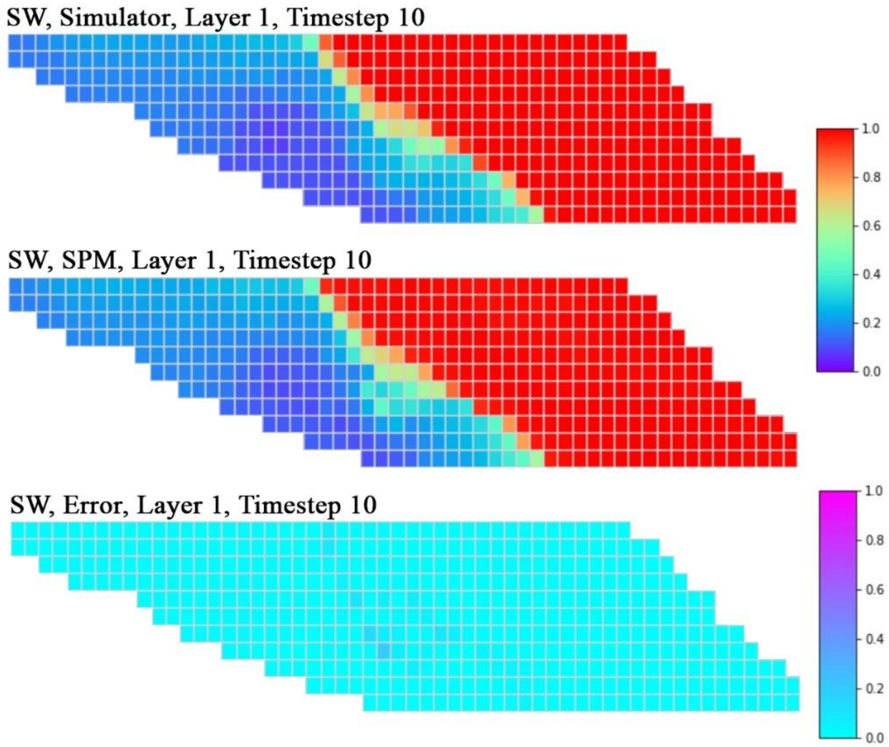


Figure 77. Grids water saturation in L1 and T10 (May 2008) for the blind test (WAG case)

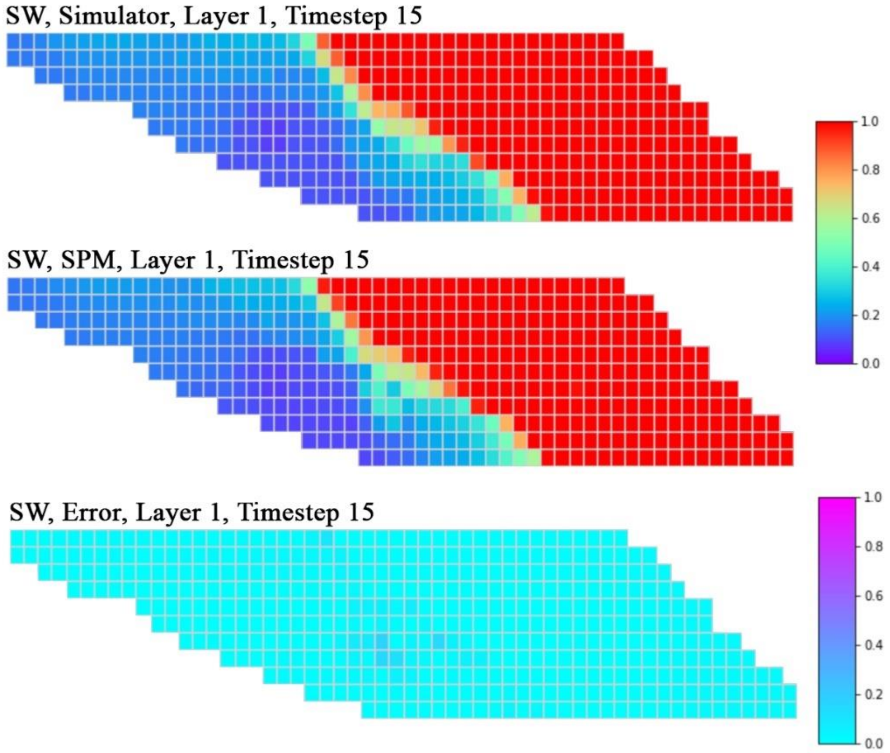


Figure 78. Grids water saturation in L1 and T15 (Jan 2010) for the blind test (WAG case)

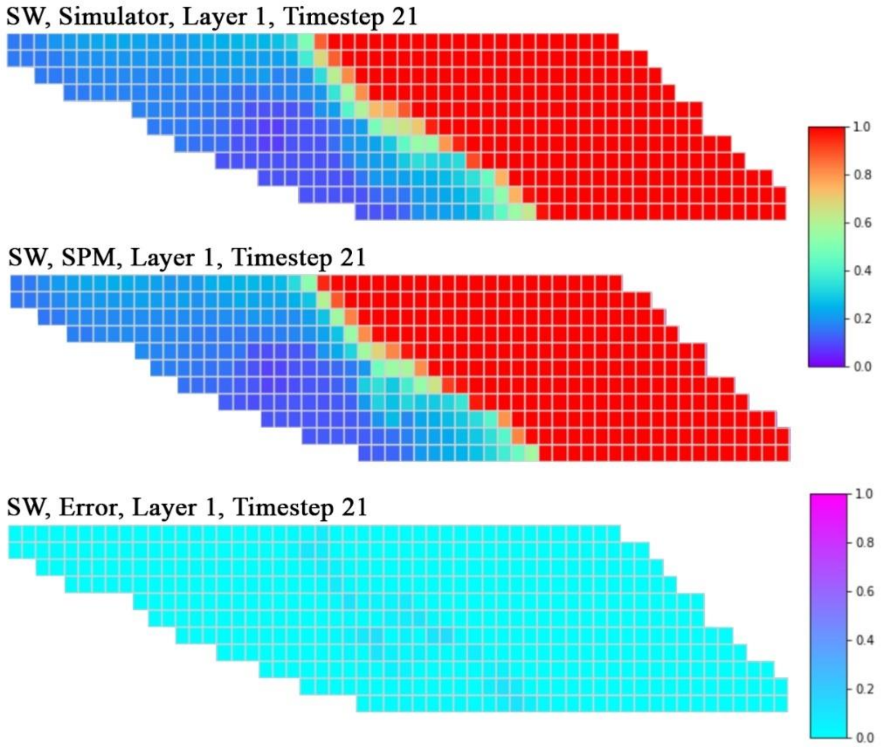
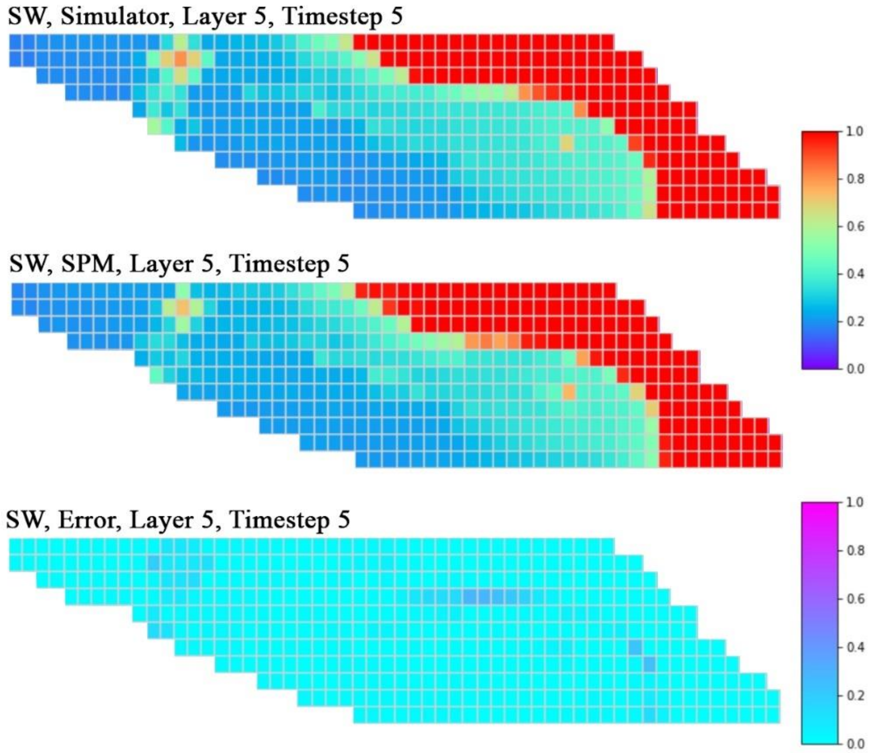


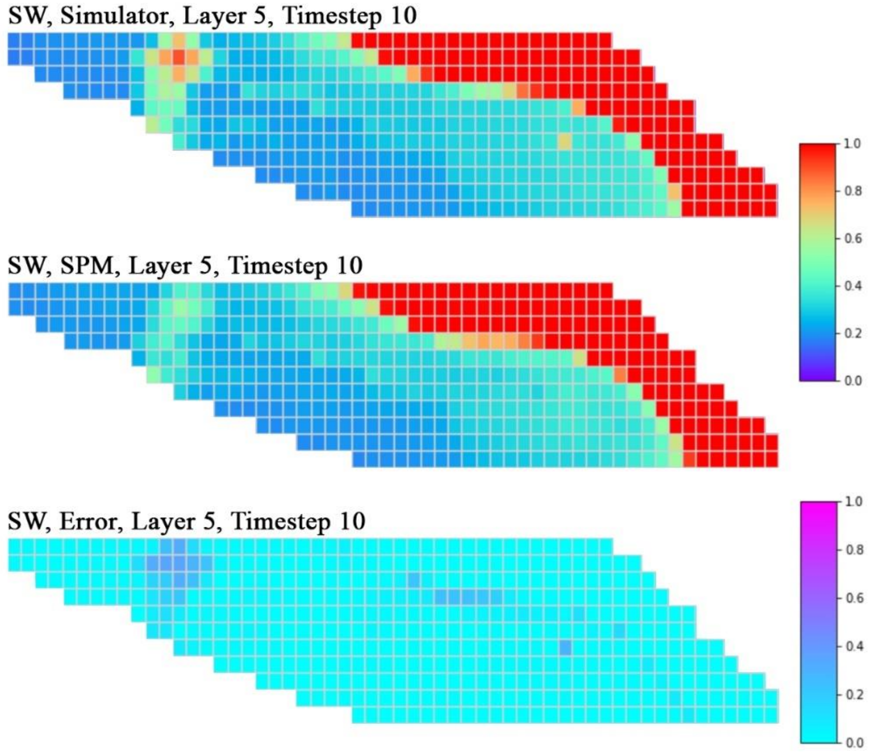
Figure 79. Grids water saturation in L1 and T21 (Jan 2012) for the blind test (WAG case)


SW, Simulator, Layer 5, Timestep 5

SW, SPM, Layer 5, Timestep 5

SW, Error, Layer 5, Timestep 5

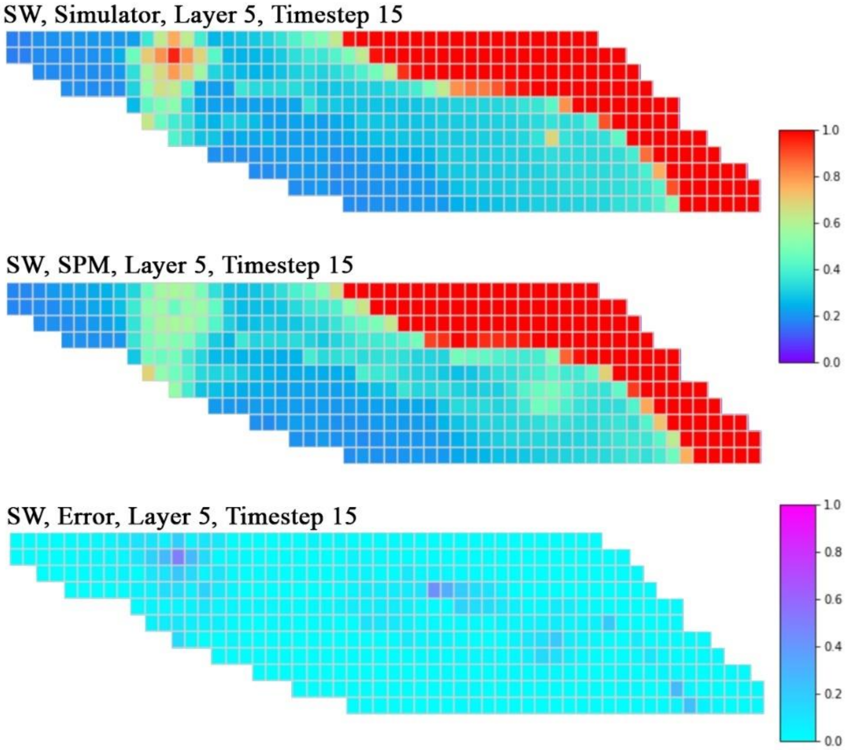
Figure 80. Grids water saturation in L5 and T5 (Sept 2006) for the blind test (WAG case)


SW, Simulator, Layer 5, Timestep 10

SW, SPM, Layer 5, Timestep 10

SW, Error, Layer 5, Timestep 10

Figure 81. Grids water saturation in L5 and T10 (May 2008) for the blind test (WAG case)

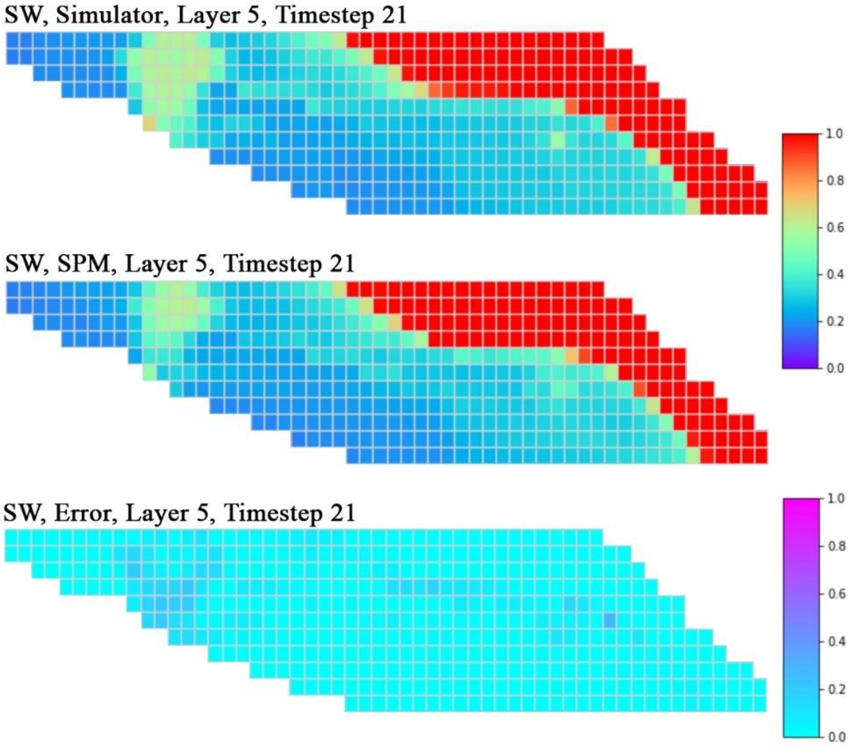


SW, Simulator, Layer 5, Timestep 15

SW, SPM, Layer 5, Timestep 15

SW, Error, Layer 5, Timestep 15

Figure 82. Grids water saturation in L5 and T15 (Jan 2010) for the blind test (WAG case)

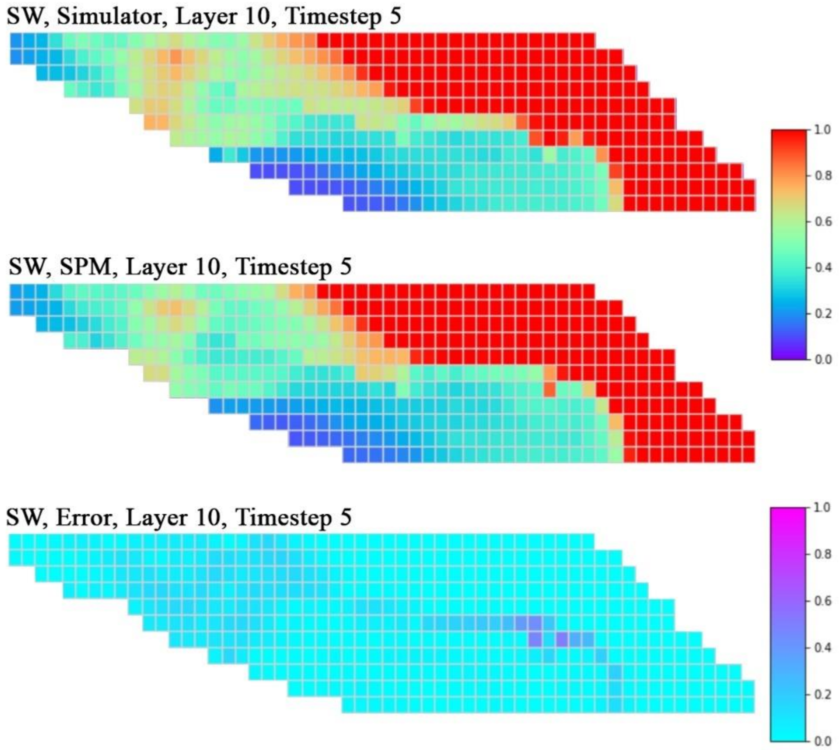


SW, Simulator, Layer 5, Timestep 21

SW, SPM, Layer 5, Timestep 21

SW, Error, Layer 5, Timestep 21

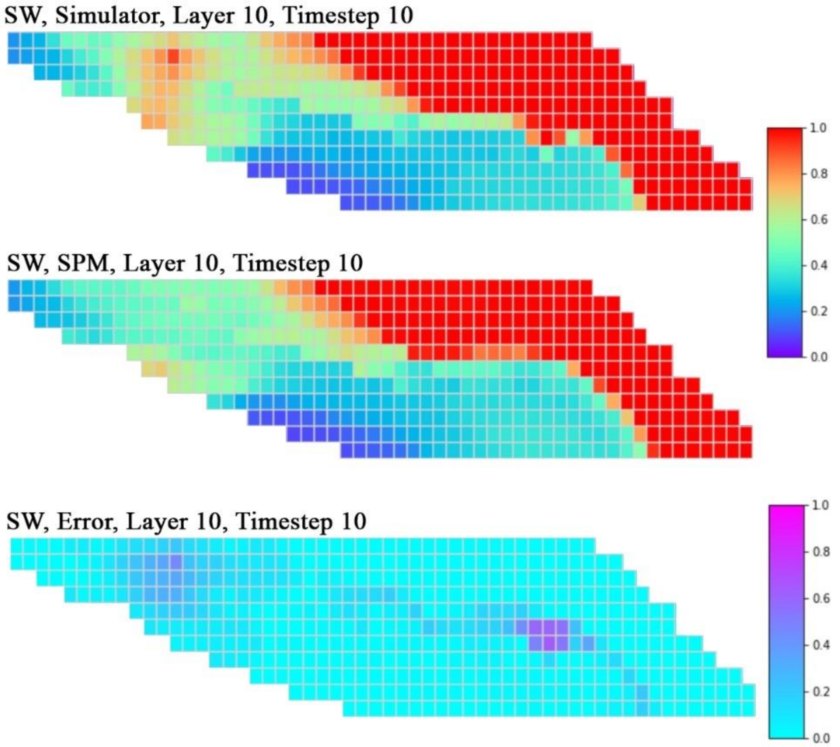
Figure 83. Grids water saturation in L5 and T21 (Jan 2012) for the blind test (WAG case)


SW, Simulator, Layer 10, Timestep 5

SW, SPM, Layer 10, Timestep 5

SW, Error, Layer 10, Timestep 5

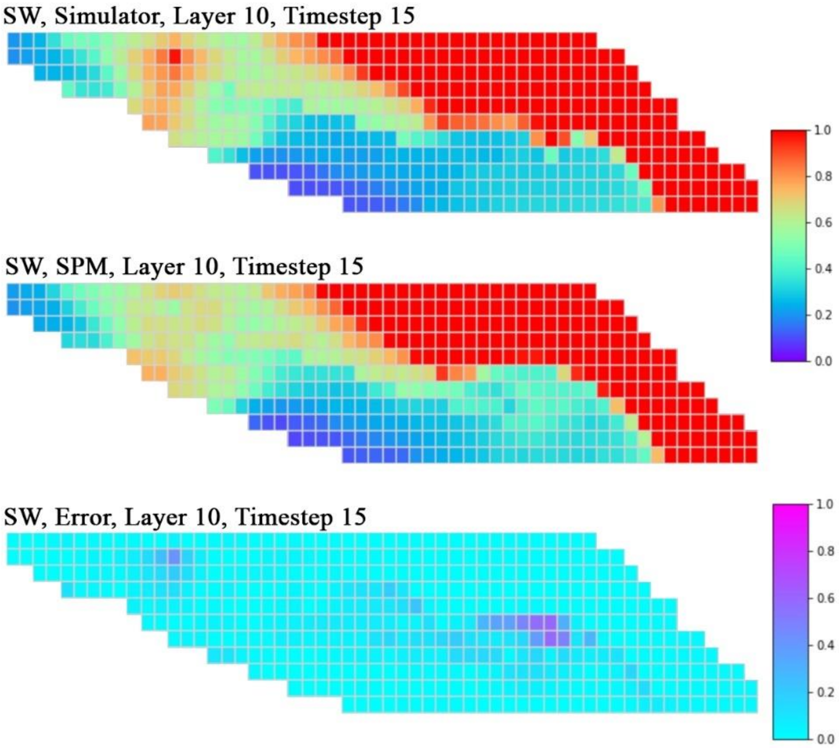
Figure 84. Grids water saturation in L10 and T5 (Sept 2006) for the blind test (WAG case)


SW, Simulator, Layer 10, Timestep 10

SW, SPM, Layer 10, Timestep 10

SW, Error, Layer 10, Timestep 10

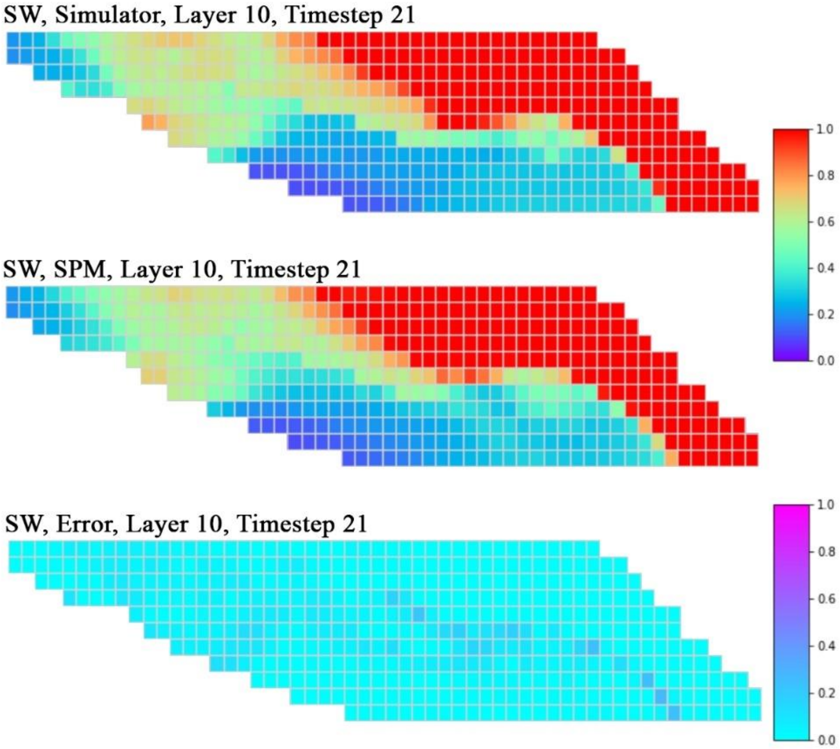
Figure 85. Grids water saturation in L10 and T10 (May 2008) for the blind test (WAG case)


SW, Simulator, Layer 10, Timestep 15

SW, SPM, Layer 10, Timestep 15

SW, Error, Layer 10, Timestep 15

Figure 86. Grids water saturation in L10 and T15 (Jan 2010) for the blind test (WAG case)


SW, Simulator, Layer 10, Timestep 21

SW, SPM, Layer 10, Timestep 21

SW, Error, Layer 10, Timestep 21

Figure 87. Grids water saturation in L10 and T21 (Jan 2012) for the blind test (WAG case)

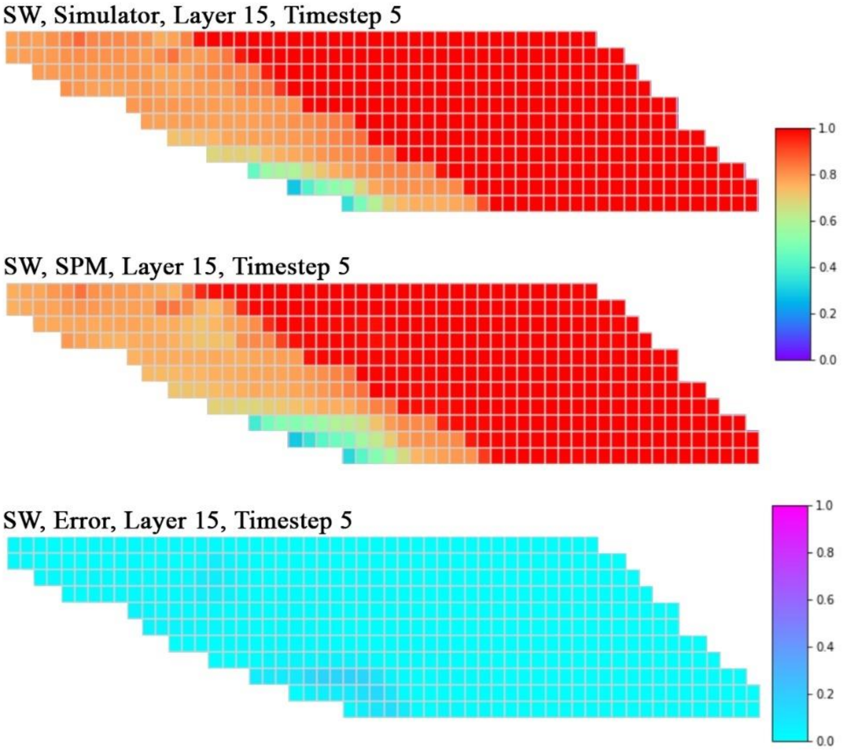


Figure 88. Grids water saturation in L15 and T5 (Sept 2006) for the blind test (WAG case)

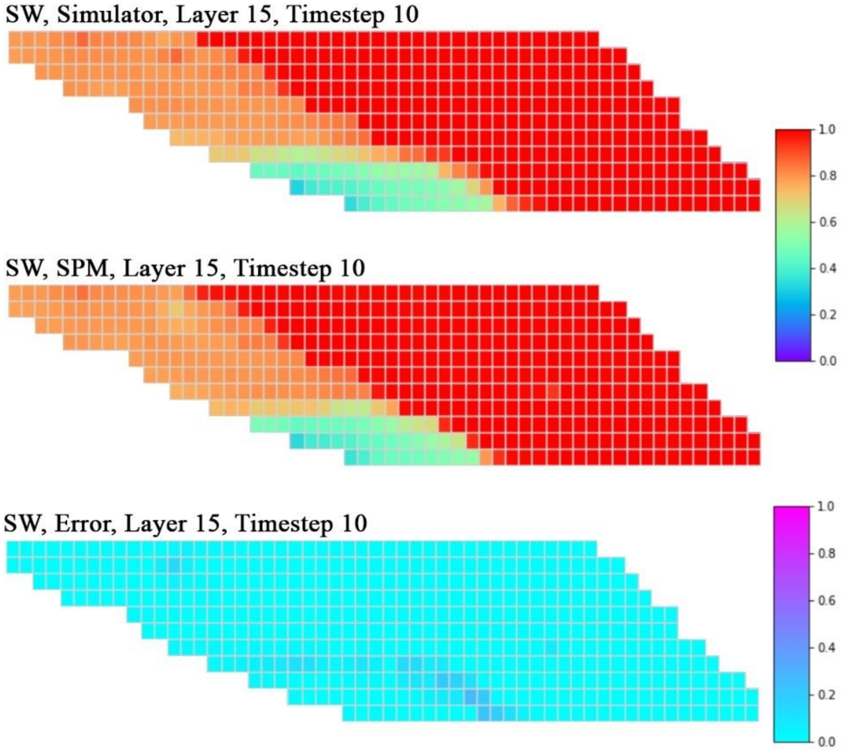


Figure 89. Grids water saturation in L15 and T10 (May 2008) for the blind test (WAG case)

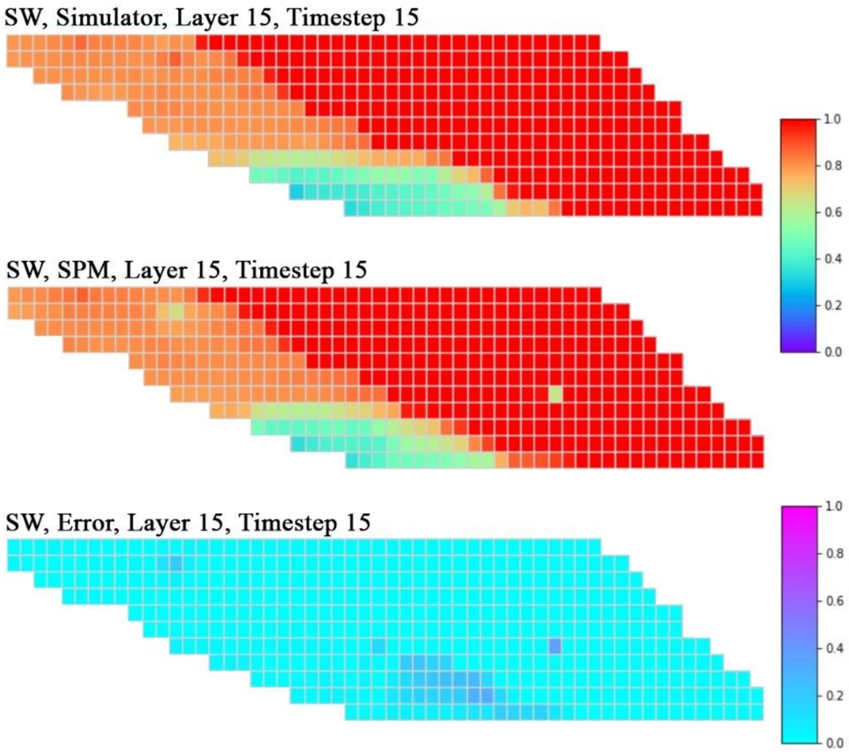


Figure 90. Grids water saturation in L15 and T15 (Jan 2010) for the blind test (WAG case)

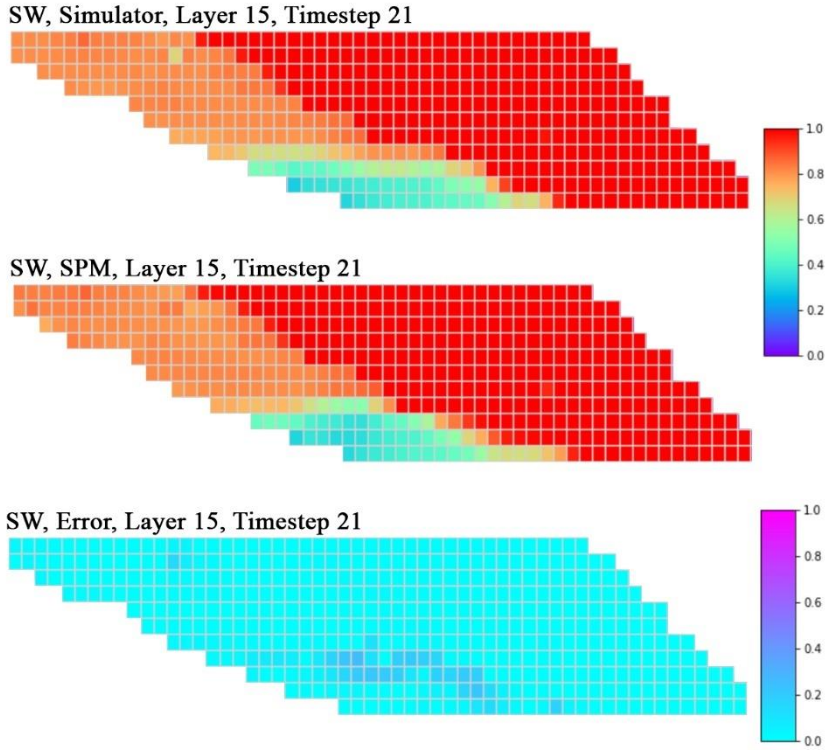
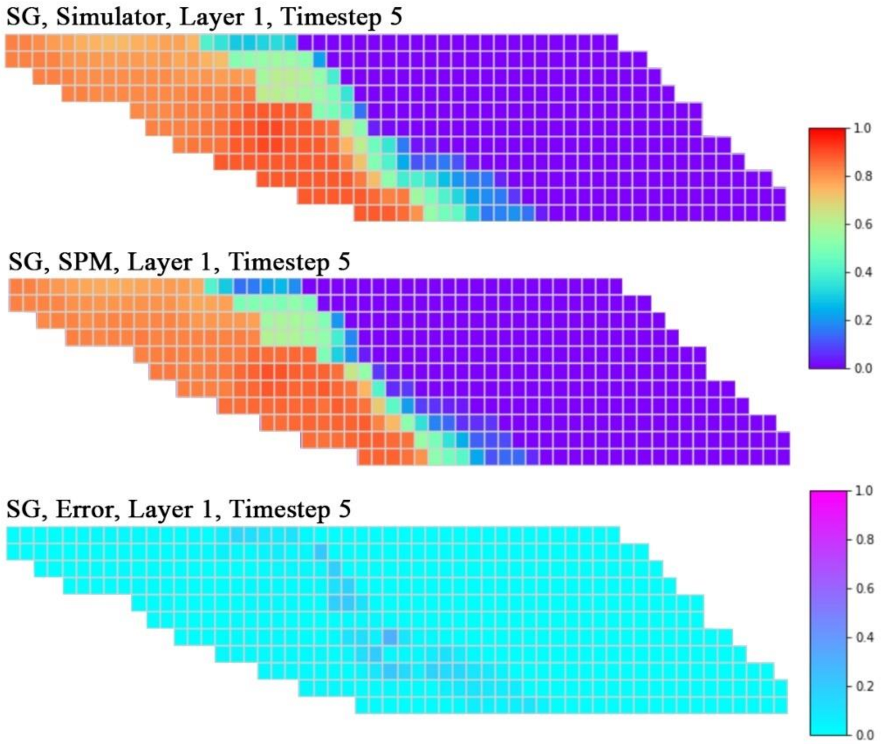


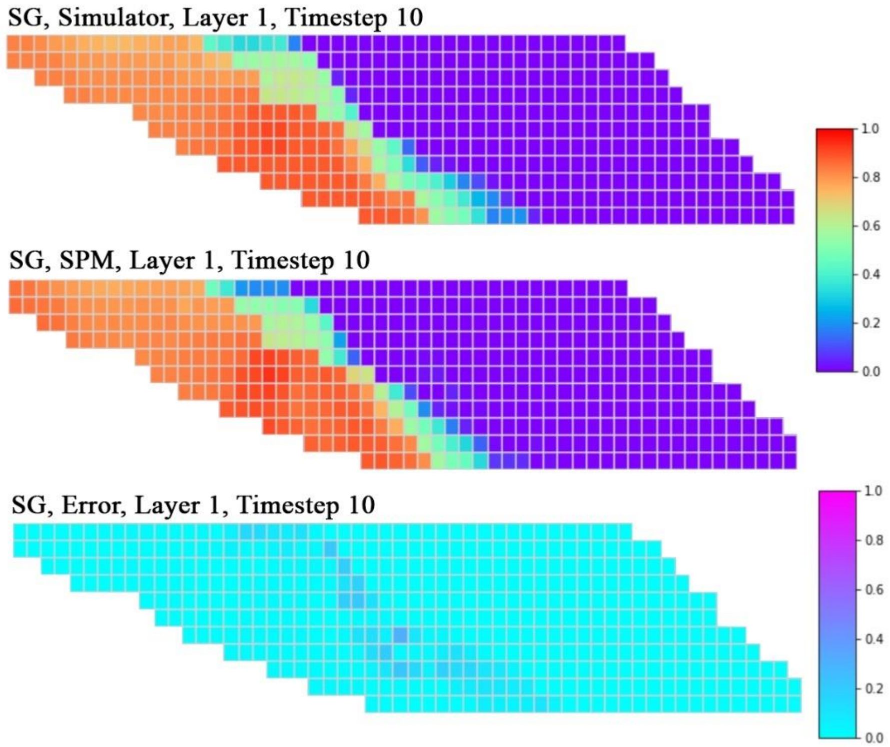
Figure 91. Grids water saturation in L15 and T21 (Jan 2012) for the blind test (WAG case)


SG, Simulator, Layer 1, Timestep 5

SG, SPM, Layer 1, Timestep 5

SG, Error, Layer 1, Timestep 5

Figure 92. Grids gas saturation in L1 and T5 (Sept 2006) for the blind test (WAG case)


SG, Simulator, Layer 1, Timestep 10

SG, SPM, Layer 1, Timestep 10

SG, Error, Layer 1, Timestep 10

Figure 93. Grids gas saturation in L1 and T10 (May 2008) for the blind test (WAG case)

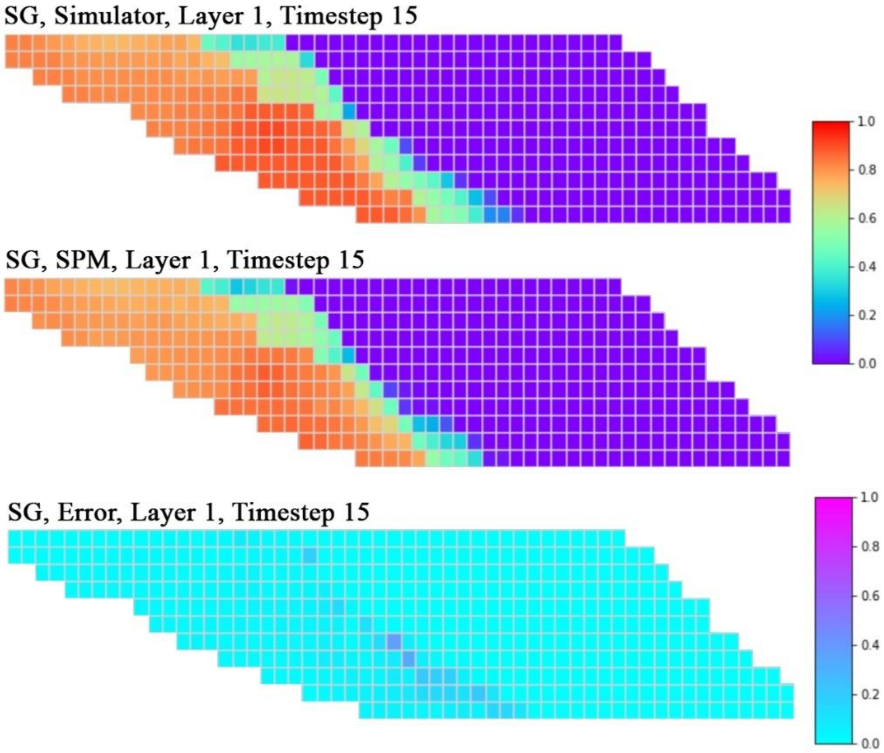


Figure 94. Grids gas saturation in L1 and T15 (Jan 2010) for the blind test (WAG case)

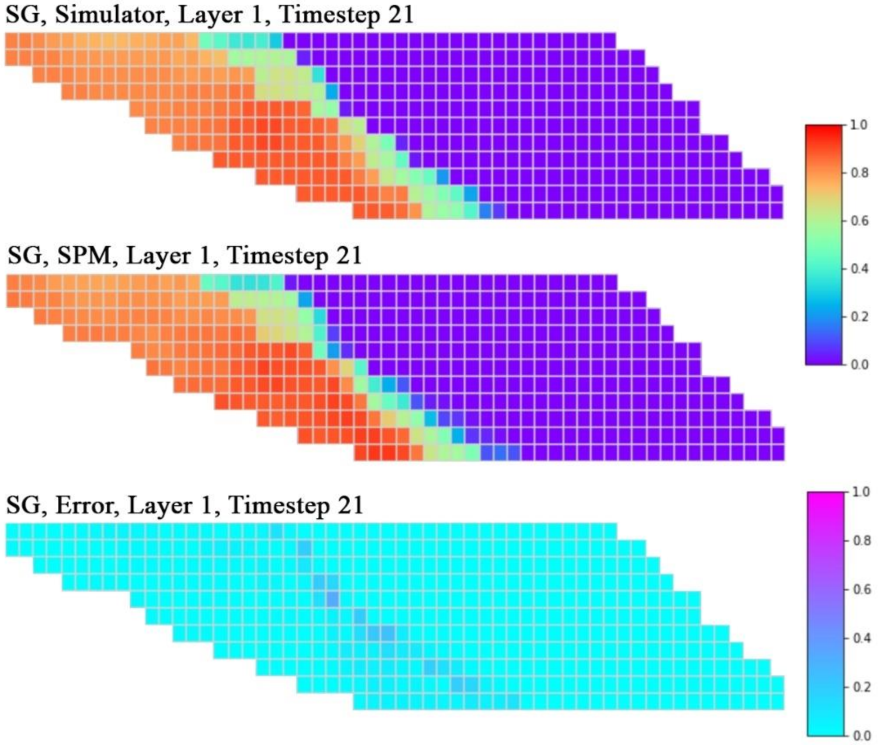


Figure 95. Grids gas saturation in L1 and T21 (Jan 2012) for the blind test (WAG case)

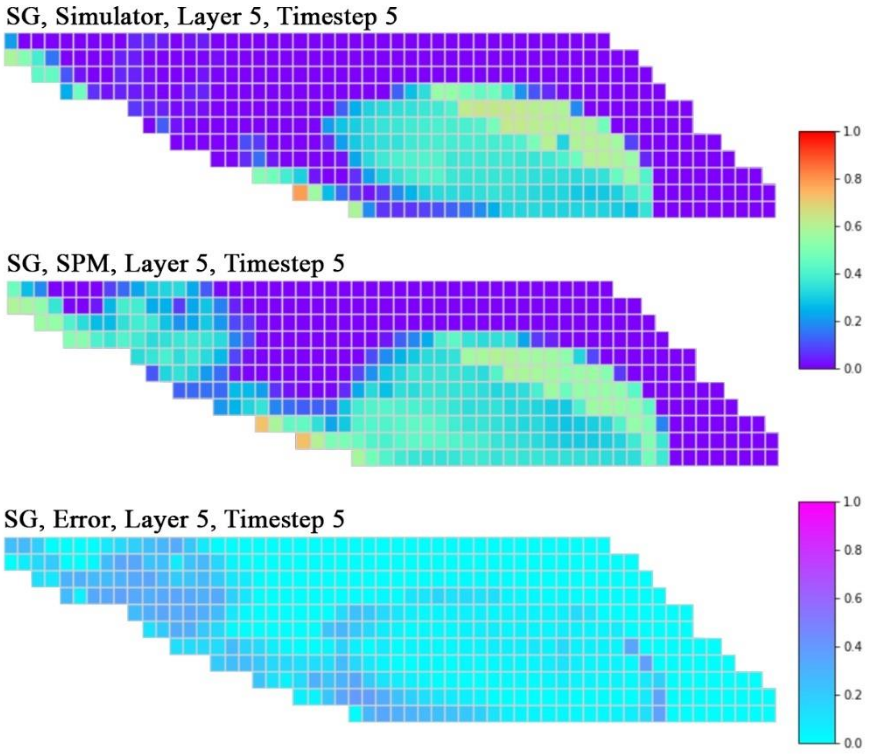


Figure 96. Grids gas saturation in L5 and T5 (Sept 2006) for the blind test (WAG case)

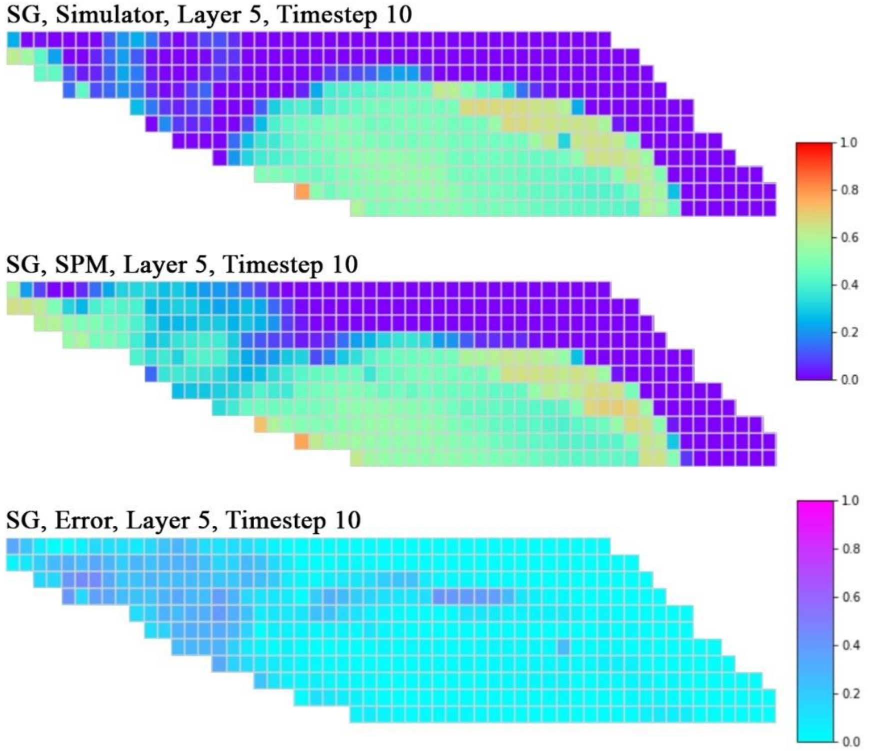


Figure 97. Grids gas saturation in L5 and T10 (May 2008) for the blind test (WAG case)

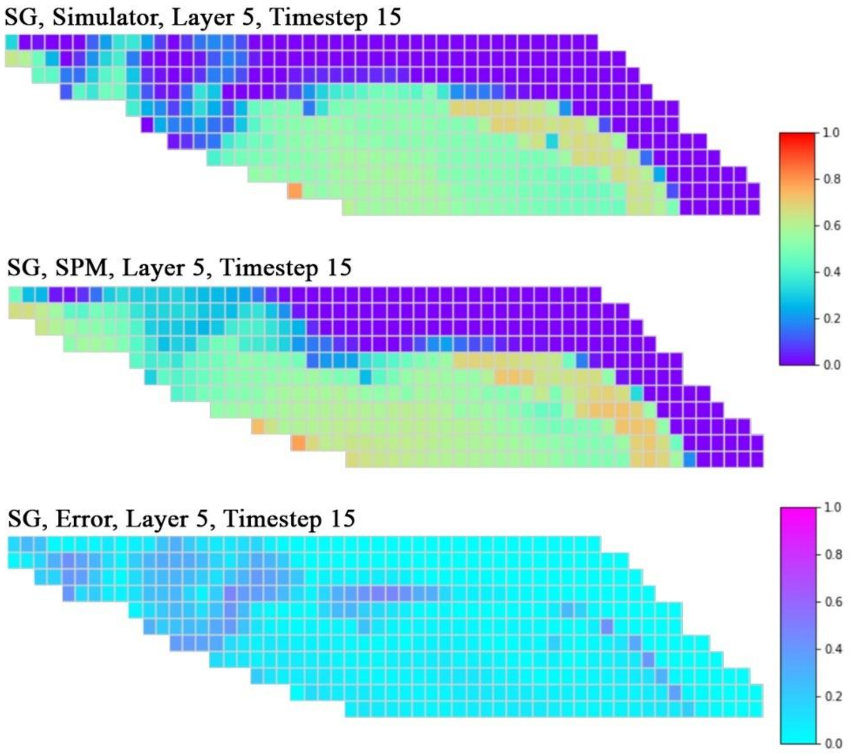


Figure 98. Grids gas saturation in L5 and T15 (Jan 2010) for the blind test (WAG case)

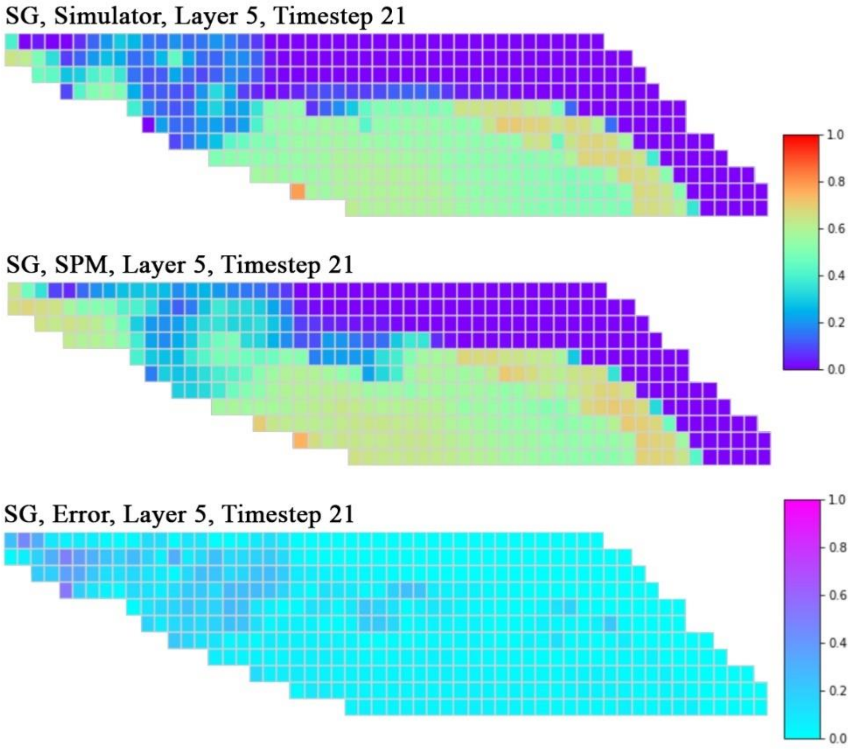


Figure 99. Grids gas saturation in L5 and T21 (Jan 2012) for the blind test (WAG case)

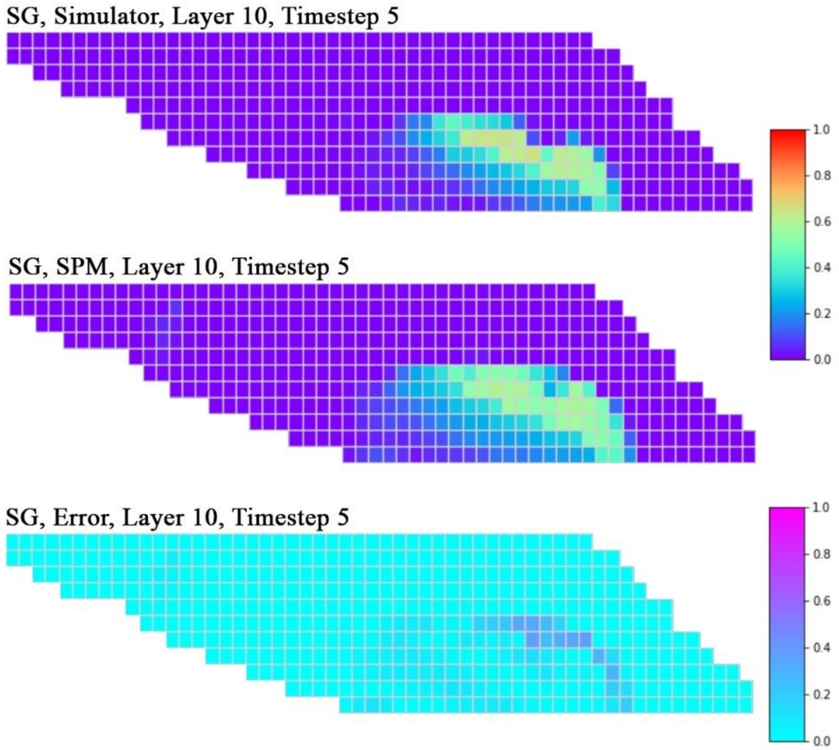


Figure 100. Grids gas saturation in L10 and T5 (Sept 2006) for the blind test (WAG case)

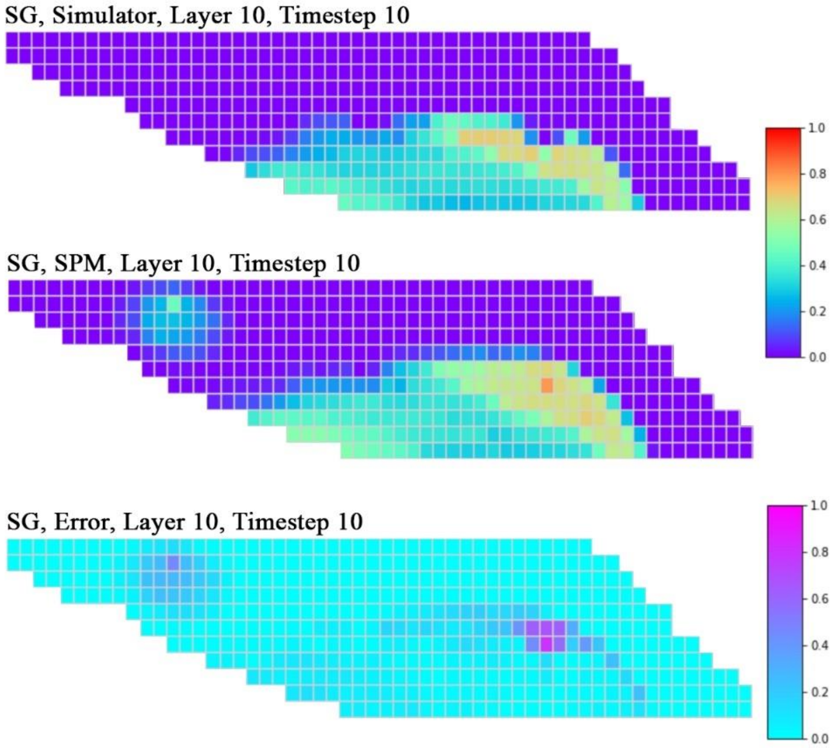


Figure 101. Grids gas saturation in L10 and T10 (May 2008) for the blind test (WAG case)

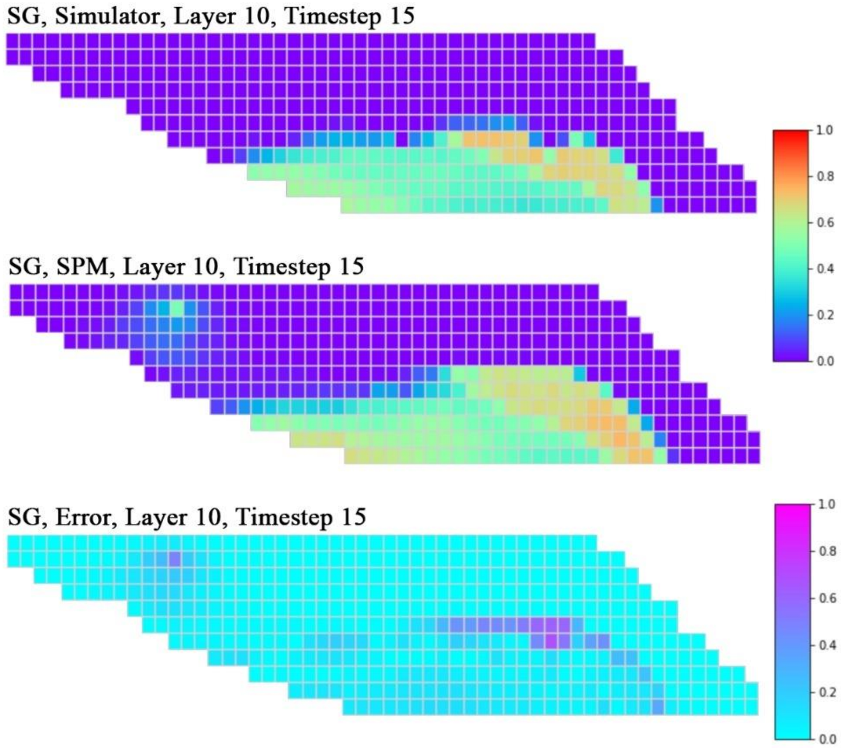


Figure 102. Grids gas saturation in L10 and T15 (Jan 2010) for the blind test (WAG case)

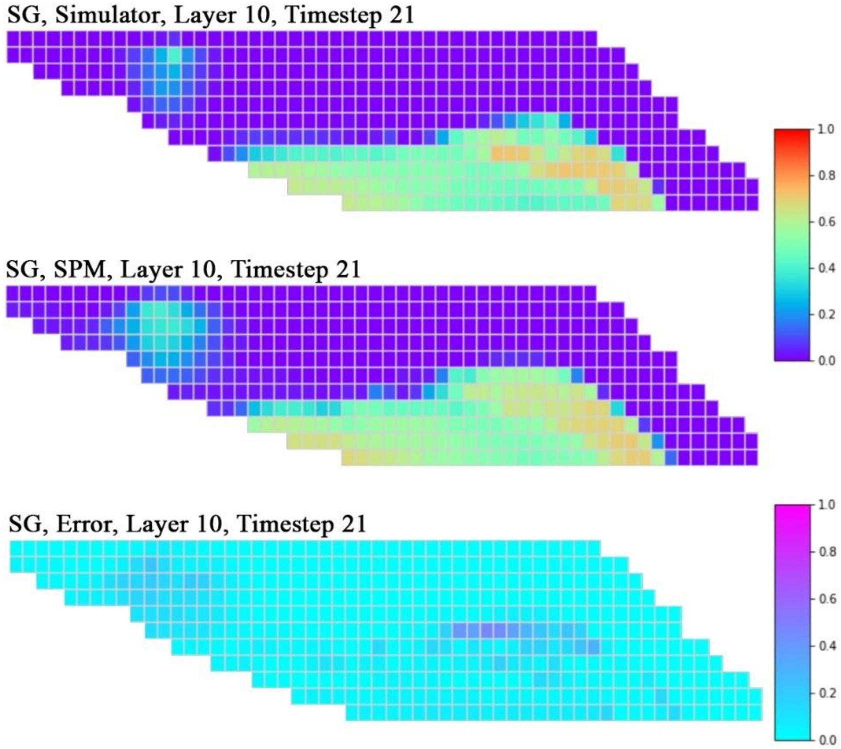


Figure 103. Grids gas saturation in L10 and T21 (Jan 2012) for the blind test (WAG case)

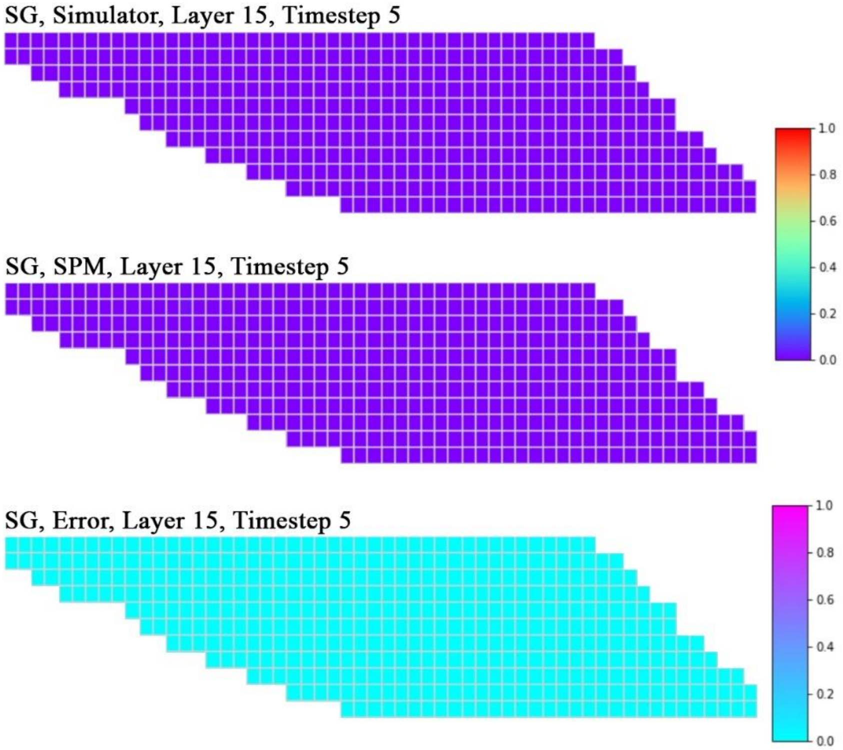


Figure 104. Grids gas saturation in L15 and T5 (Sept 2006) for the blind test (WAG case)

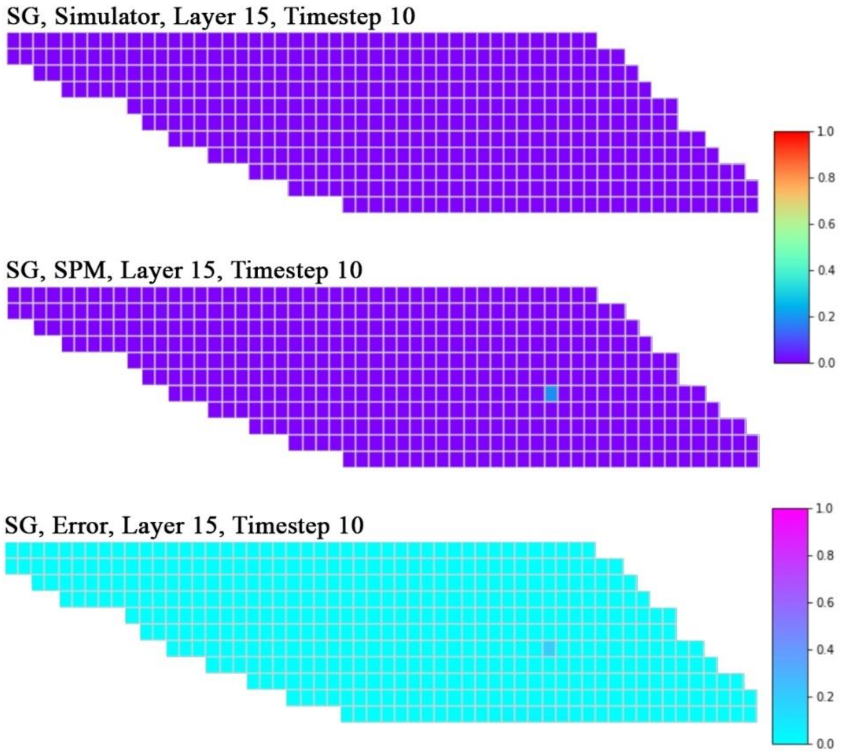


Figure 105. Grids gas saturation in L15 and T10 (May 2008) for the blind test (WAG case)

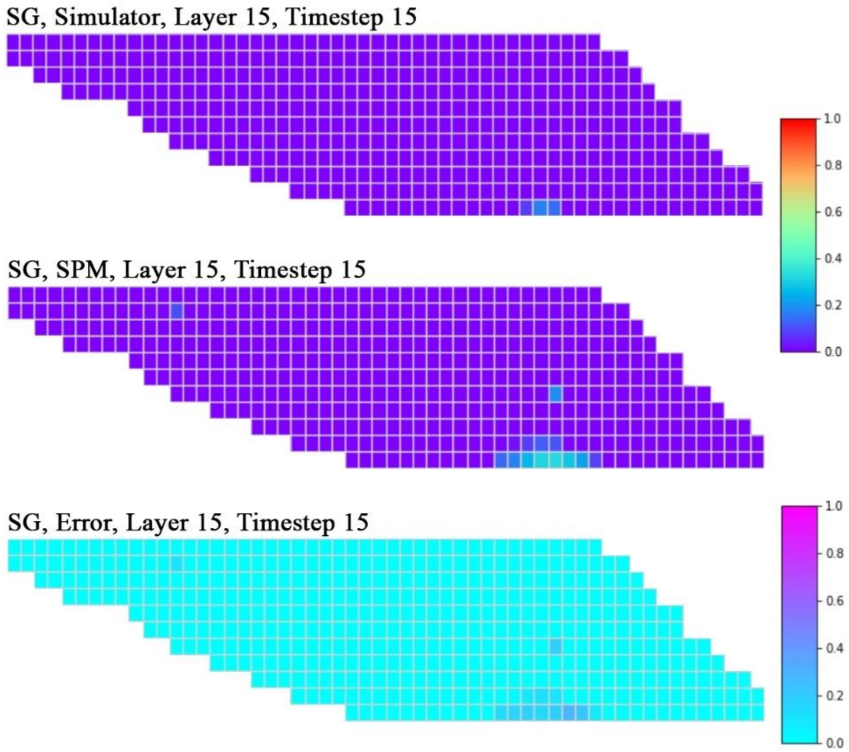


Figure 106. Grids gas saturation in L15 and T15 (Jan 2010) for the blind test (WAG case)

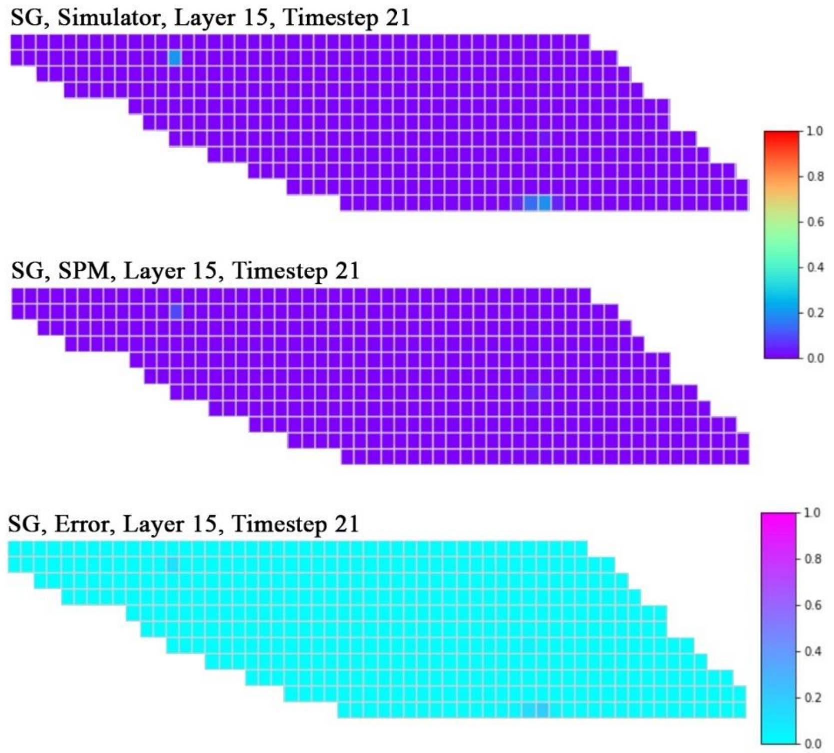


Figure 107. Grids gas saturation in L15 and T21 (Jan 2012) for the blind test (WAG case)