ESAIM: PROCEEDINGS AND SURVEYS, June 2025, Vol. 81, p. 145-167
Didier Auroux, Martin Campos Pinto, Bruno Després, Victorita Dolean, Stéphane Lanteri, Victor Michel-Dansac

PREDICTING ULTIMATE HYDROGEN PRODUCTION AND RESIDUAL VOLUME DURING CYCLIC UNDERGROUND HYDROGEN STORAGE IN POROUS MEDIA USING MACHINE LEARNING

RAYMOND MUSHABE[1], JEAN DONALD MINOUGOU[2], DAVID LANDA-MARBÁN[3], BIRANE KANE[3] AND TOR HARALD SANDVE[3]

Abstract. Underground hydrogen storage (UHS) is a promising solution for balancing renewable energy supply by storing excess energy as molecular hydrogen (H_2) for later use. Large-scale UHS in geological formations requires reservoir simulations of cyclic loading scenarios to optimize storage operations. Such simulations must account for complex physical phenomena, including reservoir flow dynamics, trapping mechanisms, and reactions with minerals and bacteria. Fast H_2 injection and production operations generate extensive datasets, necessitating high computational power to capture intrinsic temporal and spatial variations for reliable predictions. Machine learning (ML) offers a viable approach to addressing these computational challenges while improving operational efficiency and ultimately cost reduction. This study explores how ML models trained on simulated UHS data in porous media can predict ultimate hydrogen production and trapped hydrogen due to factors such as hysteresis, dissolution, and capillary pressure. The OPM Flow reservoir simulator was used to generate cyclic field data, which was used to train time-series neural network (NN) models. The models were fine tuned through hyperparameter optimization and cross-validated before deployment. Results indicated that ML models effectively predicted hydrogen production and residual storage with high accuracy, as measured by mean squared error (MSE) and mean absolute error (MAE). A key finding was that ML-based predictions significantly reduced computational time. In one reservoir-scale case, ML reduced computation time by 6773% compared to OPM Flow simulations on a four-layer reservoir model. Even greater accuracy and efficiency were achieved using a one-layer model with a horizontal well under a complex cyclic schedule. By capturing complex physical uncertainties, ML provides a powerful complement to traditional reservoir simulation methods. This research contributes to UHS development by demonstrating the potential of ML for optimizing storage operations and enabling fast, data-driven decision-making.

1. INTRODUCTION

The increasing demand for clean energy has prompted extensive research into the development of energy storage technologies that can cope with the imbalance between the supply and demand of renewable energy [19,42]. Hydrogen as an environmentally friendly and versatile energy carrier has become a favorite option in the energy transition. One of the major options for large-scale H_2 storage is underground storage in porous media. Hydrogen storage in geological porous media has several advantages. First, H_2 can be stored for the long term,

[1] University of Bergen; e-mail: ramus8091@uib.no
[2] University of Stavanger; e-mail: wendpanga.j.minougou@uis.no
[3] NORCE Research AS; e-mail: dmar@norceresearch.no, bika@norceresearch.no & tosa@norceresearch.no

Article published online by EDP Sciences and available at https://www.esaim-proc.org or https://doi.org/10.1051/proc/202581145

allowing production and injection cyclically, matching seasonal variation in energy demand [16]. Secondly, hydrogen having a high energy density by weight can be stored in significant volumes thanks to the large capacity offered by depleted gas reservoirs and aquifers [18]. This in turn provides an option for long-term energy security. In addition, depleted hydrocarbon fields are well characterized after many years of operation and would not require astronomical expenditures for reservoir characterization for H_2 storage. The existing infrastructure in such fields can be repurposed for H_2 storage. Finally, underground/subsurface hydrogen storage (UHS) provides additional safety benefits compared to surface storage since the risks of probable explosions or leaks are minimized [29]. Cyclic operation of hydrogen storage can also facilitate the integration of intermittent renewable energy sources into the energy grid. The integration is achieved by storing the excess electricity generated during periods of surplus and delivering it in periods of high demand [33].

Machine learning techniques have attracted attention in recent years thanks to their contribution to solving complex engineering issues in the energy sector. Their domain of application spanned from photovoltaic systems [21] to oil prices prediction [2] through to probabilistic forecasting of wind power [40], and building energy forecast [44]. Machine learning techniques have shown great prediction performance when applied to studies related to carbon capture and storage [2,7,28]. Furthermore, studies have attempted to predict the inter-facial tension (IFT) between different gases and the brine system with more or less accuracy [6,27,28,46]. The recent study by [34] shows how deep learning methods can be used to predict contact angles in brine-hydrocarbon systems.

Noticeable steps have been made towards the prediction of physical parameters linked with the interaction between brine and hydrogen [30]. However, the implication and effectiveness of machine learning prediction methods to hydrogen recovery and residual volumes in the context of cyclic operations remains to be proven. Parameters such as recovery factor and residual saturations are important for effective reservoir management and storage optimization purposes. As seen in [14], multiple cyclic injections of CO_2 could result in a reduced effective permeability leading to an increase in residual trapping. Numerical simulations as well as field observations have shown that injectivity is influenced by multiple cycles of injection [17,36,38]. It is known from [26] that relative permeability hysteresis is a key factor that influences the residual hydrogen saturation during cyclic loading and unloading of H_2 reservoir system.

Understanding the effect of such fluid flow parameters on the efficiency of UHS operations can help to reduce the uncertainty spectrum, historically associated with cyclic loading and unloading of the reservoir system. To reduce the uncertainty level but also for a more realistic forecast, field-scale reservoir simulations are normally used to study fluid flow, distribution, and recovery parameters. Unfortunately, geological reservoir systems typically have complex petrophysical and thermodynamic parameters that could make numerical simulations of any random phenomena challenging in terms of computational cost, speed, and accuracy [17,26,31,45]. When we consider that simulations of fast cyclic operations are needed to render underground/subsurface hydrogen storage technology viable, the challenge at hand becomes even more demanding.

With the large amount of data expected to be generated in such operations, the predicting power of machine learning algorithms can be leveraged to provide cheap, fast, and effective ways to forecast crucial parameters. The impact of machine learning in the UHS industry can be quite effective when we consider the large amount of data that needs to be generated as well as the large amounts of uncertainties linked to reservoir physics and simulations. For example, in the case of a ternary schedule of either injection, storage, or production over a one-year cyclic operation, the number of possible scenarios will be equivalent to 3^{52} (6.4610819E+24). A one-time task can potentially be quite expensive from the computational point of view but also hard to predict using traditional reservoir simulators typically designed for the oil and gas industry operations. It is therefore in our interest to look for machine learning (ML) models that scale down computational complexity while offering accuracy in a short time without affecting the results [4,5,43]. Such a task is systematically illustrated in this paper, in which a sequential *TensorFlow/Keras* neural network (NN) model is applied. TensorFlow as a machine learning system operates on a large scale for various platform environments [3] including its integration in Keras. Through Tensorflow, a wide range of tools, libraries, and community resources can be leveraged to build and train various types of machine-learning models. Furthermore, its integration with Keras provides an intuitive and user-friendly Python API, which makes it more applicable in executing ML tasks. Keras' modularity and extensibility make it an extractive platform [20] and thus ML models can indeed be defined as sequences of layers, making it possible to create complex neural network architectures. The extensibility also allows users to customize the neural network layers, and create loss functions and other metrics.

The application of TensorFlow/Keras models [1, 13] in subsurface energy storage prediction exists in the literature. Using the Smeaheia region in Norway as a case study, [8] assessed seismic monitoring using simulations and statistical analysis. They performed simulations of reservoir properties and generated synthetic seismic data which were used to fit a convolutional NN model to predict the probabilities of leaking or sealing fault in an aquifer. The implementation of the convolutional NN was done using Keras with *RELUs* as activation functions and a batch size of 16. Additionally, [39] developed an artificial NN model using Keras methods to quantitatively predict the effectiveness of trapping mechanisms in saline aquifers. The model was constructed using an optimal number of neurons as well as hidden layers and applied to an aquifer of Pohang Basin in South Korea. A deep learning inversion training model based on the *U-net* architecture, built in Python using Keras library was used by [41] to establish a mapping relationship between seismic data and CO_2 saturation. The training aimed to estimate CO_2 saturation as a function of subsurface location. Recently [37] integrated a workflow using OPM Flow and Keras to train neural networks in fine-scale simulations to be used in coarser grids. The application is on near-well modelling, where the computations of well indices are performed evaluating the NN internally in the simulator.

In this paper, several cases of cyclic hydrogen operations are simulated by considering reservoir physics and geometry to generate synthetic data that could be fed to a Keras model to predict critical parameters such as the recovery factor (RF) and residual H_2 amount. This study broadly **aims** to show the potential of machine learning in predicting short-cycle UHS operations. During the operations, the simulated data can be enriched/replaced with real field data to fully utilize the machine learning potential. The **main target** is to compare the predictions from simulation runs against those from the machine-learned algorithm, to use both interchangeably during UHS. The described ML steps are conducted on the data sets from the simulation runs. Therefore, numerical and computational errors in the simulated data sets are assumed to not be transferred to the ML training and testing rounds. To achieve the set targets, the work in this paper is split into four parts:

(1) Definition of the reservoir models and simulation scenarios
(2) Data processing, machine learning model choice, model training, validation, and testing
(3) Prediction evaluation on injection and production data sets
(4) Conclusion and future outlook

The data sets upon which the ML models are trained were generated using an open-source reservoir-scale simulator *flow*. This simulator is developed by The open porous media initiative [35]. The initiative consists of many modules including a fully implicit finite volume black-oil reservoir simulator amongst others. The workflow builds on existing frameworks and libraries such as DUNE ([9, 12, 15]) and C++ Libraries. The corresponding source code contains modules interacting with each other for a task at hand and offers a fully implicit discretization of a black-oil type model in line with industry standards [35]. *ResInsight* post processor is used as a 3D visualization tool and and it is also part of the OPM initiative.

2. Methodology

2.1. Key mathematical models

OPM Flow is an open-source simulator for subsurface applications such as hydrocarbon recovery, CO_2 storage, and H_2 storage [35]. The H_2 storage functionality is enabled by adding the H2STORE keyword on the input data deck to the simulator. The implemented mathematical model for hydrogen storage is based on models found in the literature. Here we describe the main equations and provide the references for more details in the formulation. The fluid filling the pore space in the reservoir consists of two components and two phases, i.e., the water and hydrogen components ($i \in \{\mathrm{H_2O}, \mathrm{H_2}\}$) and the gaseous and liquid phases ($\alpha \in \{g, l\}$). The two-phase Darcy's law for phase α is:

$$\mathbf{u}_\alpha = -\frac{k_{r,\alpha}}{\mu_\alpha}\overline{\overline{K}}\left(\nabla p_\alpha - \rho_\alpha \mathbf{g}\right), \tag{1}$$

where $\mathbf{u}_\alpha$ is the flux discharge per unit area [m/s], $k_{r,\alpha}$ the relative permeability [-], μ_α the dynamic viscosity [Pa·s], p_α the pressure [Pa], ρ_α the density [kg/m^3], $\mathbf{g}$ the gravity (9.81 [m/s^2]), and $\overline{\overline{K}}$ a symmetric tensor of rank 2 for the rock permeability [m^2]. The component mass conservation is written as:

$$\sum_{\alpha=g,l}\left[\frac{\partial}{\partial t}\left(\rho_\alpha \phi s_\alpha \chi_\alpha^i\right) + \nabla\cdot\rho_\alpha\left(\mathbf{u}_\alpha \chi_\alpha^i - s_\alpha \phi D_\alpha \nabla \chi_\alpha^i\right)\right] + q^i = 0, \quad i \in \{\mathrm{H_2O}, \mathrm{H_2}\} \tag{2}$$

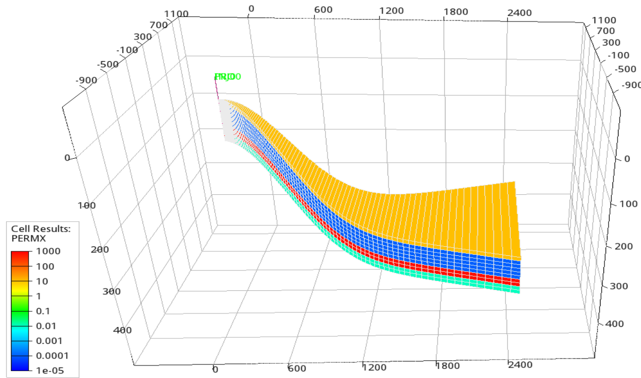
where ϕ is the porosity [-], s_α the saturation [-], χ_α^i the component mass fraction in phase α [-], D_α the molecular diffusion [m^2/s], and q^i the source/sink term [kg/(s m^3)]. The phase partitioning of each component is given by the equality of the fugacities between the two phases. The fugacities are given as specified in [25]. In the examples considered herein vaporization of water into the gas-phase is neglected. The phase saturations, pressures, and components fulfill the following conditions:

$$p_g - p_l = p_c, \quad s_l + s_g = 1, \quad \text{and} \quad \sum_{i=\mathrm{H_2O},\mathrm{H_2}} \chi_\alpha^i = 1, \quad \alpha \in \{g, l\}, \tag{3}$$

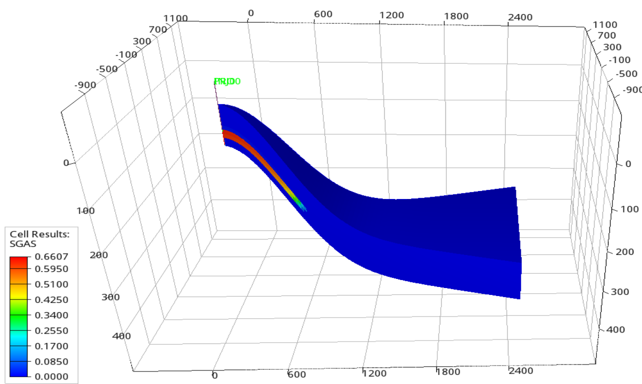
where $p_c(s_l)$ is the capillary pressure [Pa]. The relative permeabilities and capillary pressure are given as input tables as a function of the gas saturation. In this paper, we use the Brooks-Corey functions, where these relationships are given as a function of the effective saturations s_α^*:

$$s_\alpha^* = \frac{s_\alpha - s_{\alpha,i}}{1 - s_{l,i} - s_{g,i}}, \quad k_{r,\alpha} = k_\alpha (s_\alpha^*)^{n_\alpha}, \quad \text{and} \quad p_c = p_e (s_l^*)^{-\frac{1}{n}}, \tag{4}$$

where $s_{\alpha,i}$ is the saturation below which the phase is immobile [-], k_α the end-point scaling [-], p_e the entry pressure [Pa], and n_α and n fitting coefficients [-]. In addition, we include hysteresis in the gas relative permeability, using the Killough model [22]. The PVT (pressure, volume, temperature) properties such as densities, viscosities, and diffusivity as a function of pressure, temperature, and composition are computed internally by using analytical correlations and models from the literature [11]. We refer to the OPM Flow manual for details on these models [10]. OPM Flow uses industry-standard I/O formats, which are usually generated by commercial software (e.g., the grid, injection schedule, rock properties, saturation functions). Here, we use pyopmnearwell [24], an open-source Python-based framework to simulate near-well dynamics using OPM Flow. The framework allows to choose and define the application to model (e.g., hydrogen storage), define the grid dimensions and refinement, set the different rock and fluid properties, define the location of the wells, and set the schedule for the operation of wells. In addition, it includes post-processing routines to generate figures comparing results for different runs. The setting of the simulations is via a single configuration file, which allows for reproducibility of results.



(A) The four layers in the grid model are differentiated by absolute permeability values in the horizontal direction. The highest permeability layer is indicated in red.



(B) Gas saturation map in the storage layer at one point in time during cyclic loading and unloading. No-layer cross-flow is observed during the injection cycles.

FIGURE 1. One half of the Gaussian grid model described by [31] is used to avoid redundant simulation data but also halves on the simulation time for each run.

2.2. Data simulation and visualization

In this work, the *OPM Flow* simulator is utilized to simulate and generate data from different scenarios of injection, storage, and production while considering parameters such as economic limit, hysteresis, and reservoir geometry. The simulations are accordingly conducted on published reservoir models [31,45] and the scheduling is based on scenarios that depict real field-scale operations. The simulations are categorized in two based on the grid reservoir model used. In each category, a base case scenario is simulated first followed by its counterpart complex scenario. Complexity is defined by the fluid flow physics captured in the simulation model or the operations schedule followed.

The first geological model shown in figure 1, is based on the benchmark study performed in [31]. The reservoir is 2500 m in the radial direction and 100 m in the vertical direction with 100 and 10 cells in each of the directions respectively. The initial reservoir pressure is 4 MPa and with a homogeneous temperature of 50°C. It has four layers of variable thickness and rock properties. Table 1 shows the different layer properties within the grid model. The base case simulation scenario on this geological model **(base case 1)** is initiated by first injecting H_2 in the reservoir for 365 days at a rate of 20 000 kg/day to work as a cushion gas for pressure maintenance reasons. The injection phase is followed by 90 days of storage to allow for field-pressure stabilization. The third part of the operation consists of 27 possible cases of a 70-day storage scenario inspired by [31]. For each possible case, different combinations of operations are simulated. That implied that each cycle could either be injection, production, or storage as shown in figure 2. Different configuration files, linked to the simulator are used to create such a ternary schedule that mimics all possible operational schedules for UHS. The simulation runs generated a wide range of data including the amounts of hydrogen injected, produced, and the residual gas that remains trapped in the reservoir upon recovery. In all possible production cases, the economic limit is fixed at 95%, meaning the production is stopped if water fraction exceeds 10% of the produced volume. And in all scenarios especially during injection, the well is pressure-controlled at a maximum value of 400 barg.

Layer	Thickness [m]	ϕ [-]	$K_{xx}/K_{yy}/K_{zz}$ [mD][1]	$s_{l,i}$ [-]	$s_{g,i}$ [-]	k_l [-]	k_g [-]	p_e [Pa]	n_l [-]	n_g [-]	n [-]
1	10	0.1	10	0.1	0.1	0.8	0.2	4E5	4	3.5	1.2
2	50	0.1	1E-4	0.1	0.1	0.8	0.2	4E6	4	3.5	1.2
3	20	0.25	700.15	0.1	0.1	0.8	0.2	4E5	4	3.5	1.2
4	25	0.1	1E-2	0.1	0.1	0.8	0.2	4E6	4	3.5	1.2

[1] 1 mD = 9.869233E-16 m^2

Table 1. Reservoir properties in grid model inspired from [31].

This geological model also captures the gravity effect on the recovered H_2 volume upon production since different grid cells are defined at different depths. In addition, capillary pressure and H_2 dissolution in the water phase contributed to the residual hydrogen volume in the reservoir upon production in the base case scenario. The first target scenario, named **target scenario 1** on this defined Gaussian-like model (figure 1) accounted for relative permeability hysteresis during injection (imbibition) and production (drainage) cycles as an additional parameter to contribute to the residual hydrogen volume. From the results of our simulations, not presented in this paper, relative permeability hysteresis and production rate economic limit plays a major role in determining hydrogen production and residual amounts. By imposing an economic limit of 95%, the amount of hydrogen being produced is reduced by almost 50% since the production well shut in whenever the hydrogen production rate is less than 95% due to pressure depletion or market factors. The reduction of produced hydrogen is even more pronounced when hysteresis is considered. In this paper, hysteresis is executed with a Killough hysteresis model on the storage reservoir layer whose flow properties are also included in table 1. Additionally, relative permeability hysteresis results in unique production data trends during cyclic operations. The variations in trends is associated with differences in relative permeability values for each phase during imbibition and drainage cycles. Predicting such data trends using ML made the work conducted in the study more impacting especially considering that multi-physics simulation forecasts are typically computationally demanding.

Layer	Thickness [m]	ϕ [-]	K_{xx}/K_{yy} [mD]	K_{zz} [mD]	$s_{l,i}$ [-]	$s_{g,i}$ [-]	k_l [-]	k_g [-]	p_e [Pa]	n_l [-]	n_g [-]	n [-]	$s_{g,i}^{immb1}$ [-]	n_g^{immb1} [-]
1	100	0.15	700.15	70.015	0.2	0.05	1	1	4E5	2	2	1.2	0.3	4

[1] Irreducible gas saturation and shape exponent for the imbibition curves in the hysteresis model.

Table 2. Reservoir properties for the horizontal well simulations.

The second geological model shown in figure 3 is based on horizontal well operations inspired from [45]. The reservoir is 10 m in the horizontal direction, and 50 m in the vertical direction and has 10 and 100 grid cells in each respective direction. The initial reservoir pressure and temperature are the same as in the first geological model (4 MPa and 50°C). Table 2 summarizes the key reservoir properties. All simulations on this model followed the same schedule that mimics a seven-season scenario for four years. Injection and production

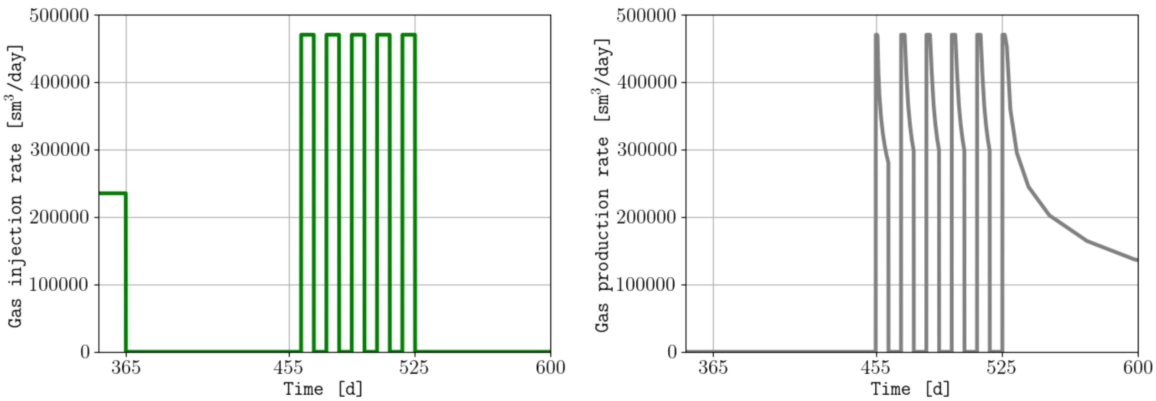
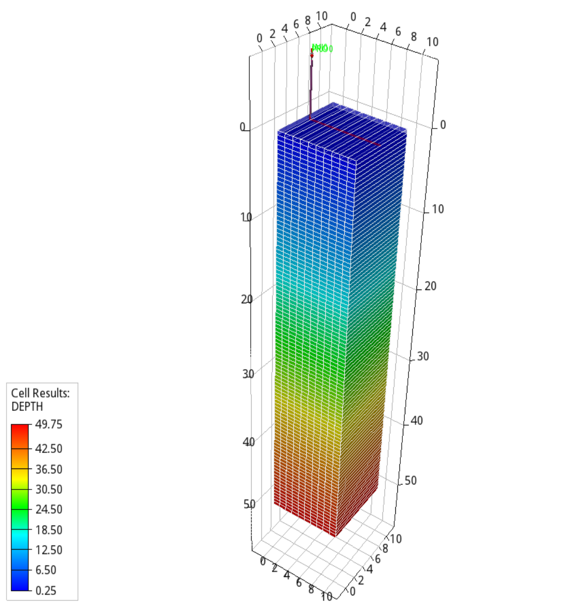
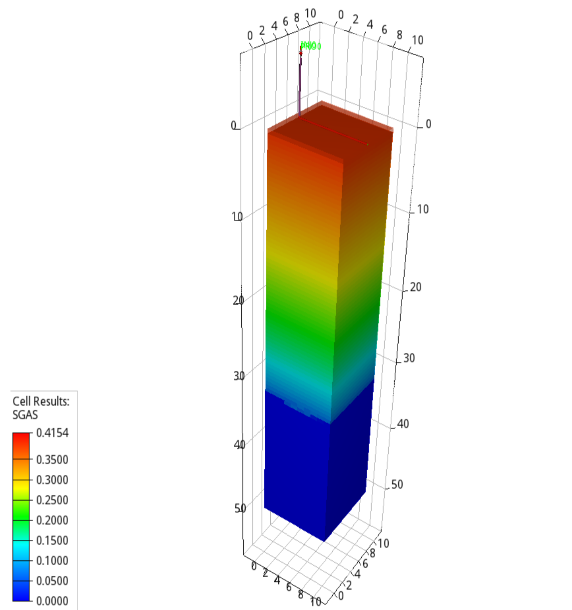


FIGURE 2. Example of injection and production rates for the storage scenario inspired by [31]. Left: The injection schedule initiated at 20 000 kg/day for 365 days and thereafter 40 000 kg/day. Right: The production cycles at 40 000 kg/day. A storage period of neither injection nor production is seen between day 365 and 455 on both plots.



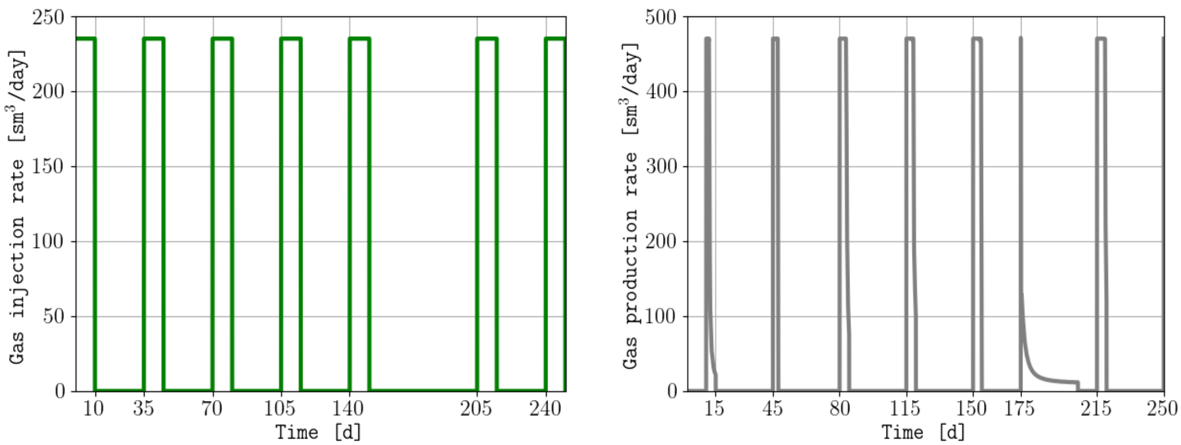
(A) Cell depth map in the model. The layer has a single permeability value.



(B) Gas saturation map mid-way the last simulation run.

FIGURE 3. The reservoir model described in [45]. In this study, homogeneity in layer permeability is considered.

schedules for the first and beginning of the second season are shown in 4a and 4b respectively. For each season, simulations are run in consecutive cycles of 10 days of H_2 injection at 20 kg/day, 5 days of production at 40 kg/day, 20 days of storage, and then 30 days of prolonged production at the end at 40 kg/day. An economic limit of 95% on the production rate is considered. The wells are pressure-controlled at 300 barg for each injection or production cycle. The base case simulations on this model, **(base case 2)** accounts for hydrogen loss due to dissolution in the water phase and capillary effects. The target scenario, **(target scenario 2)** additionally



(A) (B)

FIGURE 4. The first season and beginning of the second one for injection (left) and production (right) schedules for the seven-season cycle operations.

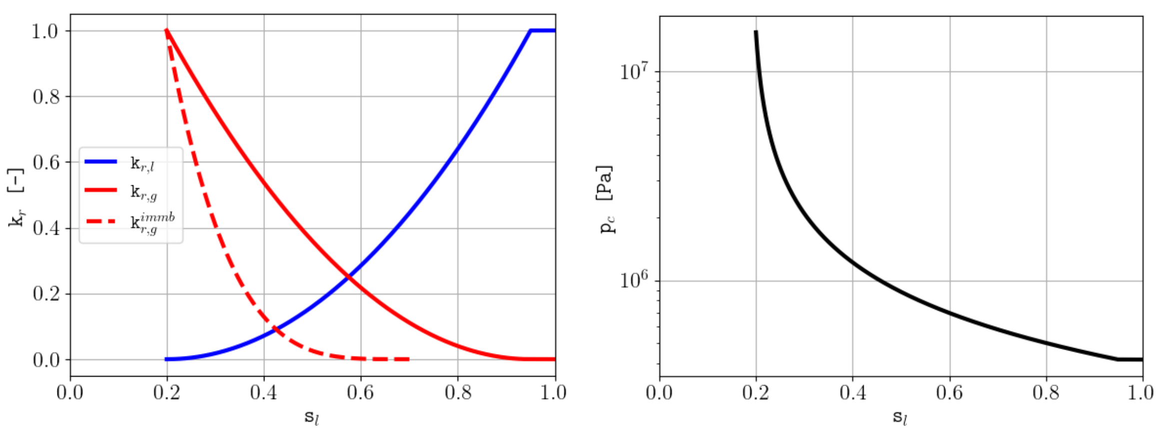


FIGURE 5. Saturation functions for the horizontal well simulations.

accounts for hydrogen loss upon production due to relative permeability hysteresis as shown by the saturation function plots in figure 5. As in target scenario 1, a Killough hysteresis model [22, 23] is used in this scenario. The objective of target scenario 2 is to capture the data trends arising due to probable complex schedules associated with UHS operations in addition to the fluid flow physics and well placement.

2.3. Machine learning Model choice and training

The data generated in both **base case 1** and **target scenario 1** is fed to a Keras time series NN to predict the volumetric parameters. Essentially the injected, produced, and residual amounts of hydrogen. The model structure consists of 6 layers in total for target scenario 1 as shown in listing 1 and only one feature with a definite look-back value of 6 is passed to the input layer. One input layer, four dense hidden layers, and one output layer. For each input sequence, each data point is assigned either 2 or 3 neurons to better learn the temporal variations in the dataset. The skeletal structure for the base case can be seen in figure 7. The model applies ReLU (rectified linear unit) as its activation function to account for non-linearity in the data

Listing 1. Model 1 NN structure

```
# A sequential model and its architecture to optimise
#
def build_model(hp):
    model = keras.Sequential()
    model.add(Dense(3, input_shape=(look_back,), activation='relu'))
    #
    # OPTIMISATION ON THE NUMBER OF NEURONS FOR EACH LAYER.
    #
    model.add(Dense(units=hp.Int('units', min_value=18, max_value=32, step=2),
    activation='relu'))
    model.add(Dense(units=hp.Int('units_2', min_value=14, max_value=32, step=2),
    activation='relu'))
    model.add(Dense(units=hp.Int('units_3', min_value=10, max_value=32, step=2),
    activation='relu'))
    model.add(Dense(units=hp.Int('units_4', min_value=4, max_value=32, step=2),
    activation='relu'))

    model.add(Dense(1))  # Output layer for regression

    model.compile(optimizer='adam', loss='mean_squared_error', metrics=['mae'])
    return model
```

sets. The loss function is represented by the mean squared error (MSE) described by equation (6) in addition to mean absolute error (MAE) applied in model hyperparameter optimisation rounds. The loss function that is minimised during testing and validation steps is MSE.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |X - \bar{X}| \tag{5}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (X - \bar{X})^2 \tag{6}$$

where **n** is the total number of entries and in this study its equal to 57 data points. **X** is the actual data value and $\bar{X}$ is its corresponding predicted value.

Adam is the choice optimizer used with a learning rate of 0.001. In deep neural networks, fully connected dense layers serve as the building blocks of the neural network and the weights initially assigned to each are optimized during training to fit the data in question. Weight optimization in our study is only limited to the hidden layers for each epoch. During model training, the number of epochs is set to 2000 for base case 1 and 1500 for target scenario 1 while the batch size is 2 for both. The rest of the hyper-parameters are summarized in table 3. The learning procedures starts with training 70% of each respective data set generated from the series of simulation in pyopmnearwell. Afterward, 30% of the data is used to validate and test the trained model. Finally, the trained and validated model is called to forecast parameters for hydrogen cyclic operations, namely the residual from which the recovery factor could be estimated. For data that is generated in **base case 2** and **target scenario 2**, the same sequential Keras times series NN model is used. The key difference arises from the number of dense layers that are considered which is reduced and set to two. The rest of the steps are as in the first scenario and NN model hyper-parameters are also summarised in table 3. During model training, a windowed time series approach with a varying look-back parameter is used in training the NN models on all data sets. The lookback parameter is set to 6 for the data obtained on a multi-layer reservoir model shown in

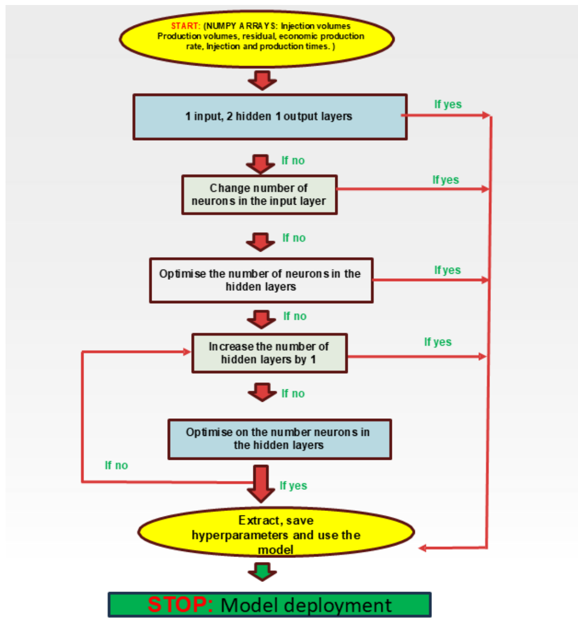


FIGURE 6. A graphical illustration of the key optimisation steps during hyperparameter tuning. The focus was on the number of hidden layers and neurons connecting them in the neural network.

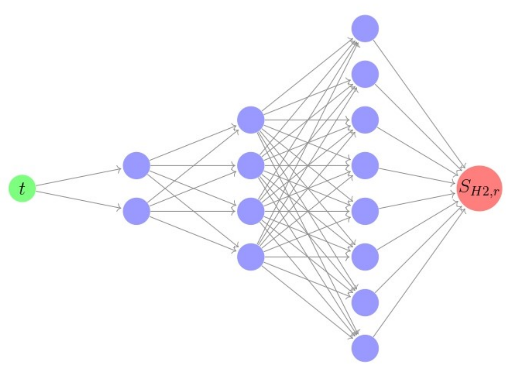


FIGURE 7. The 4-layer base case 1 NN model structure that is used to develop hyper-parameter optimization run for target scenario 1. Only one input feature is provided but each element in the layer is passed two neurons. This allows the model to learn the temporal complexity in the dataset as each neuron is provided with a unique weight or bias.

figure 1, and to either 21 (base case 2) or 42 (target scenario 2) for the single layer model shown in figure 3. All the simulation output data is extracted from summary files (.SMSPEC files) using the Ecl python tool and saved as NumPy arrays. The predefined operations schedule is used to index every dataset. A scikit-learn *MinMaxScaler* method is applied to normalize the training data set to range between 0 and 1. The validation and prediction are done on the same normalised data but the output is always converted back to the original data format using the inverse operation.

2.4. Hyperparameter tuning

A random search optimisation approach is used on the data sets that require the used baseline windowed time series sequential Keras neural network (NN) model to be optimised. The optimisation models are imported as Keras tuners and the *HyperParameter* method is fed with a dictionary of parameters to optimise as shown in listing 1. The optimisation round begins immediately after simulation data prepossessing and partitioning. The step also includes defining the baseline structure of the NN model. For two cases, the optimization is done on the number of neurons connecting the hidden layers target scenario 1. For the data simulated on the reservoir model shown in figure 3, the optimised hyper-parameters are batch size, number of hidden layers, and number of neurons. Neural network models for both base case 2 and target scenario 2 required hyper-parameter optimisation. All hyperparameter optimisation rounds followed a procedure illustrated in figure 6. The optimisation step is proceeded by fitting the optimized model on the production dataset.

CASE	Batch	Layers	Neurons	Sim time	Train-test time	Epoch	MAE	MSE	learn
Units	[-]	[-]	[-]	[s/sim]	[s/sim]	[-]	[-]	[-]	rate
Base case 1	2	4	15	64	44	2000	-	8.9E-5	0.001
Target scenario 1	2	6	60	82	68	1500	-	8.2E-4	0.001
Base case 2	21	4	22	95	72	2000	5.6E-3	5.1E-5	0.001
Target scenario 2	42	4	38	95	80	2000	6.1E-3	5.8E-5	0.001

Table 3. A summary of the key ML results and the hyperparameter values in each case study. *Sim time* refers to simulation run time

CASE	Total sim time	Total train test	Δ sim	Δ sim	Δ (sim-ML)	Δ (sim-ML)
Units	[s]	[s]	[s]	[%]	[s]	[%]
Base case 1	3648	64	0	0	3604	8190.9
Target scenario 1	4674	82	1026	28.1	4606	6773.5
Base case 2	665	72	0	0	593	823.6
Target scenario 2	665	80	0	0	585	731.2

Table 4. Comparative results between simulation runs and ML model training and validation. Δ Sim refers to difference in simulation time between base cases and target scenarios, Δ (sim-ML) is the time difference between the reservoir simulations and the machine learning computational time.

3. Results and discussion

3.1. The base case 1

Results of the **base case 1** are obtained on a reservoir model that had the same flow dynamics as summarised in table 1 both during drainage and imbibition. The only physics in the host reservoir rock accounting for hydrogen loss upon production was its dissolution in the water phase. Therefore, the effect of fluid flow and saturation history in our base system is not considered. The results are obtained on a relatively easy-to-implement schedule. When a windowed sequential Keras model with a look back factor of 6 is used, the injection trend was well captured with 70% of the training data set as seen in figure 8. The validation and testing steps on the remaining 30% also perfectly fit the data. When the model was called to predict on an injection dataset it had not seen before, the prediction was a good fit as can be seen from the value of the MSE which approaches zero. The same trend was obtained when the model was exposed to data sets obtained from the machine learning cycles. The trends achieved in this base case were expected as the volumes for and time of injections were known before being fed to the neural network model and expected not to be limited by

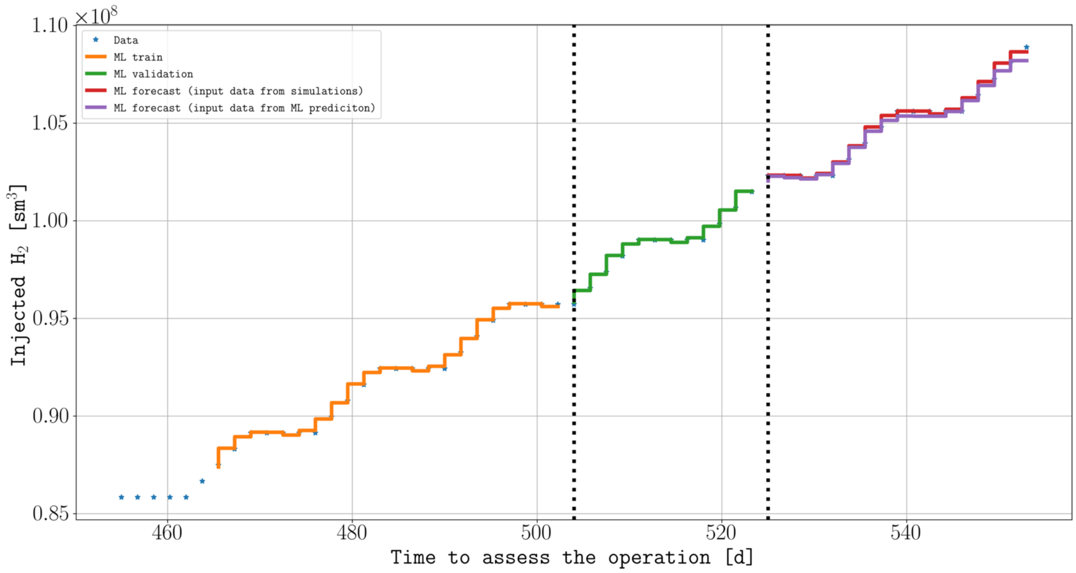


FIGURE 8. For base case 1, the injected schedule was straightforward where the cushion gas injection and prolonged storage periods were followed by equally spaced intermittent injection and production cycles. The sequential Keras time series neural network used was first tested on this data set as a checkpoint. The model was expected to perfectly fit the data points (blue stars). Any further changes to the model were bench-marked against this.

the reservoir physics. The injected volume was conditioned to be a constant value for a particular period of injection with a maximum BHP of 400 barg as the limiting boundary condition at the production well sand-face. With a working NN model on the injection data set, further training and prediction cycles are conducted on the production dataset as well as the residual volumes as shown in figure 9 and figure 10 respectively. The same NN architecture is used with two hidden layers and one input and output layer, (refer to section 4). The training, validation, and model testing steps are good, and the model captured the data trends instead of overfitting as can be seen in figure 10. The endpoint validation loss function values (MSE) between the simulated data and the predicted was about 8.9E-5, and the training and testing times lasted less than 44 seconds for all the data sets for a total of 2000 epochs. The loss function evolution during the testing and validation rounds can be seen in figure 11a, where the loss drastically drops within the first 50 epochs and stays relatively close to an acceptable range thereafter. The observed average time per epoch was about 10 ms. Forecasts on the new simulation data sets and that generated by the machine-learned algorithm are promising as seen by the purple and red curves in figure 10. At this point, the neural network model architecture used so far made good predictions based on the accuracy of the values and the relatively low computational cost in terms of run time. The graphical presentations in figure 8, figure 9, figure 10 and figure 11a support this observation. In addition, the training time of 44 seconds for 57 data points compared against a simulation time of an average of 64 seconds per simulation run (61 minutes for 57 data points) backs the observation of reduced computational cost. Further, ML manipulations were performed on a more realistic dataset that simulated scenarios intended to capture peak hydrogen market demand and supply during underground/subsurface hydrogen storage operations, and will be discussed in the next subsection. All computational tasks throughout the study were conducted on an *IdeaPad 5 pro 16" 2.5K 120 Hz Dolby atmos* local computer with an AMD RADEON graphics card.

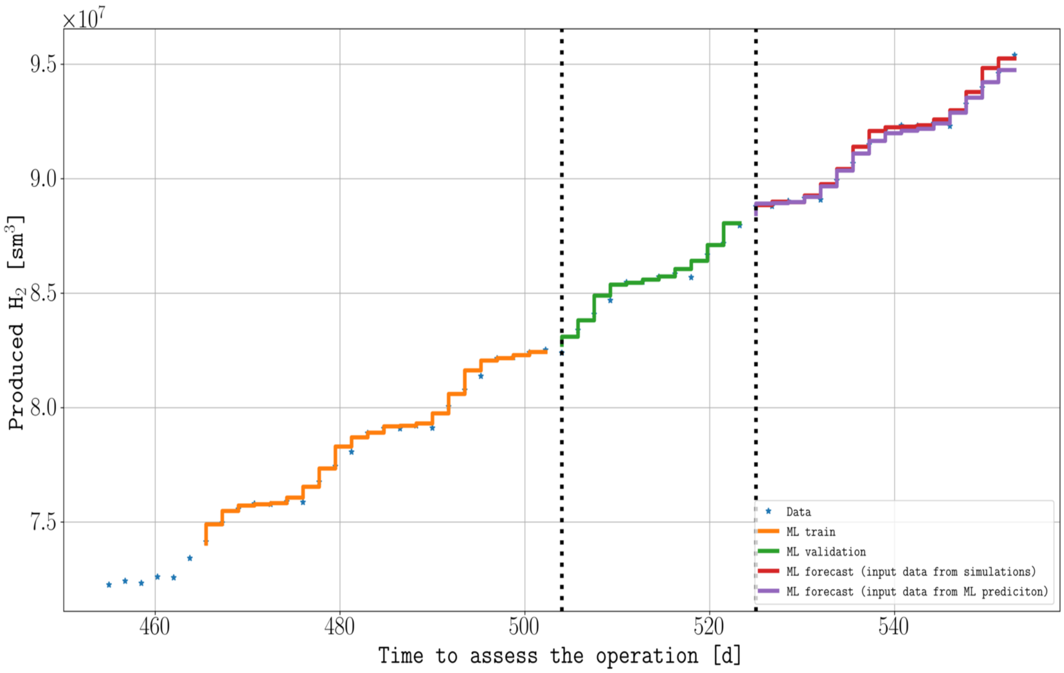


FIGURE 9. The prediction on the ultimate produced hydrogen for each production period in a base case scenario was good. The training and validation steps were good and the model did not overfit the data when it was called to predict on both the new dataset and the machine learning-generated data set as seen in the red and purple curves.

3.2. **Target scenario 1**

Results of the target scenario 1 captured underground/subsurface hydrogen storage (UHS) operations from a Gaussian grid model inspired and described in [31] and are shown in figure 12. Since the model was symmetrical, only one-half of it is used to generate the training data set and this substantially reduces the simulation run times but also avoids redundant training data. In addition to the fluid loss captured in the base case, the reservoir model in this scenario also accounted for relative permeability hysteresis both during drainage and imbibition cycles. The simulation outputs were also affected by the complexity in the schedule during loading and unloading cycles. To capture the H_2 market factors, the simulation output data was conditioned to an economic limit of 95% as described in the methodology sections. It is important to point out that the computational cost per simulation run increased from 44 seconds in the base case to about 82 seconds per simulation run. An incremental change of 28%.

The architecture of the windowed times series NN applied in the base case 1 scenario was first applied with a dataset of the same size. The neural network was first exposed to the injection dataset as an integrity check and the model fitted the data set in a similar way as was seen in the base case. However, when the cumulative production dataset is used, the training, validation, and testing steps were well executed but the model struggled to predict the new dataset. The hyper-parameters that we used in the base case are of default origin except for the number of hidden layers and neurons that are explicitly defined when designing base case 1 NN model. This implied that no hyper-parameters were optimized and therefore to make meaningful predictions,

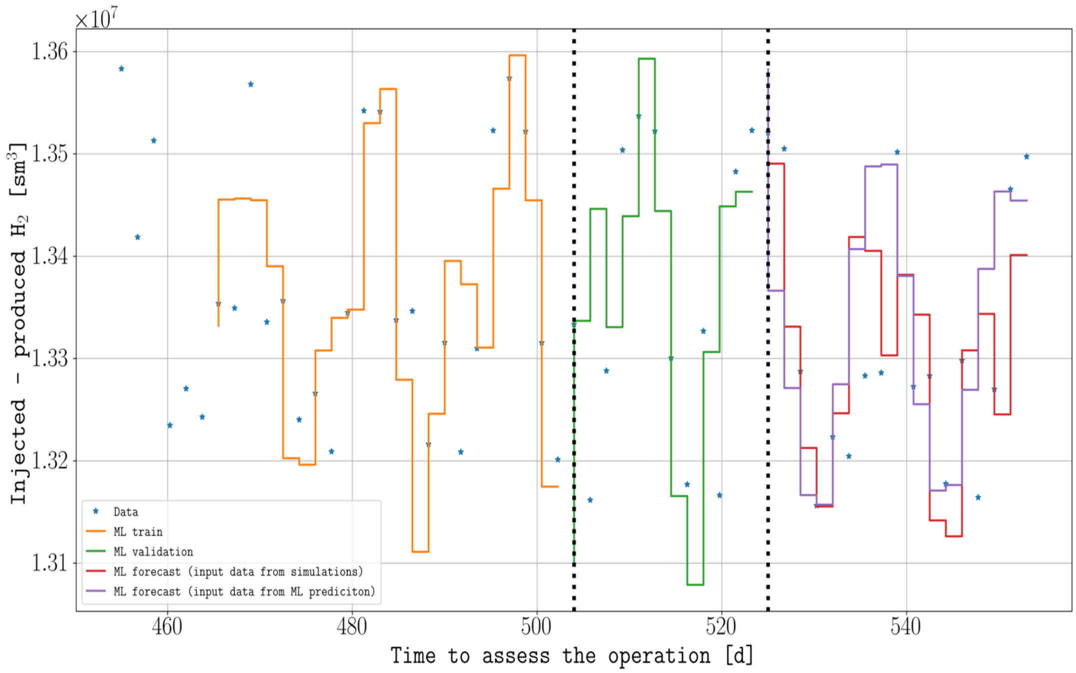
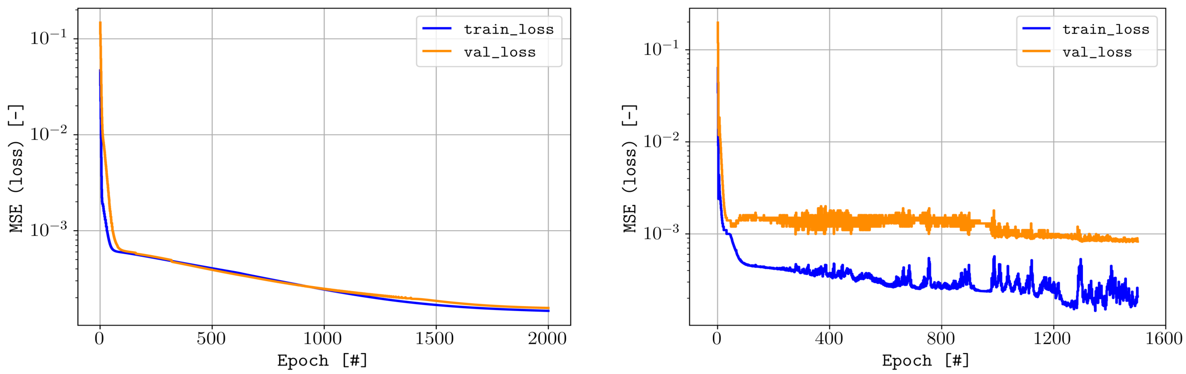


FIGURE 10. The training and validation steps on the residual data set were also good. It was even better when the model was executed on the new dataset and generated from machine learning. Note that the residual kept varying depending on the length of randomly chosen assessment times.

there was a need to conduct an optimization run. To minimize the optimization computational cost, a random optimization approach compatible with a Keras model is applied. The optimization was limited to only the number of dense layers and neurons connecting them. The objective function is the validation loss function (MAE), and the maximum number of iterations is limited to 10. The optimization step increased the number of dense layers from 2 to 4 but also substantially increased the neurons connecting them. This increased the total number of neurons in the network from 15 to 60. All the machine learning manipulations including hyperparameter optimization are conducted in one run. Conducting the optimization run first substantially improved the performance of the resultant NN model. Even with the multi-physics dataset, the cumulative training time increased by 55% to 68 seconds. With an average time of 13 ms/epoch for a total of 1500 epochs. However the final MSE during the validation steps increased to an endpoint value of 8.2E-4 in comparison with the base case values. Approximately one order of magnitude difference but still close to zero. The results of the loss functions (MSE) evolution versus epoch values during model training and validation are shown in figure 11b. In terms of computational cost, comparing the 82 seconds for training and testing an ML model against 77 minutes for all the simulation runs further amplifies the need for this technology in UHS operations. That cost difference in this case amounts to 6773.5%. The rest of the comparative performance values are summarised in table 3 and table 4. The training, validation, testing, and prediction results on the cumulative production data set using the optimized NN Keras sequential model are shown in figure 12a. The model is well trained and validated with training and validation loss values mean squared error as described before and shown in figure 11b. The model also captures data trends without necessarily overfitting during the testing and prediction rounds. However, predictions on the new simulated dataset are most of the time higher than those from machine learning results as seen by the red and purple curves respectively in figure 12a. When the same model is deployed on the residual



(A) The MSE evolution as a function of the epochs during model fitting for base case 1

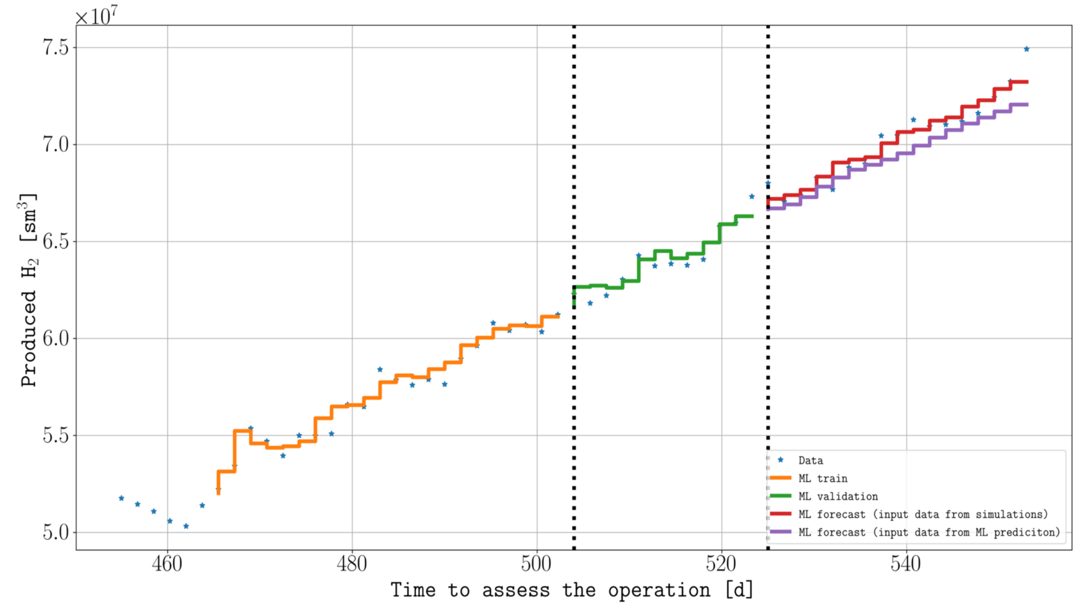
(B) The MSE evolution as a function of the epochs during model fitting for target scenario 1

FIGURE 11. The evolution behavior of the loss function (MSE) during the training and validation steps of the NN model trained on the dataset from base case 1 and target scenario 1. In both scenarios, the loss functions were monotonically declining and reduced to an acceptable window. The occurrence of the validation loss being greater than the training loss was also a positive observation since the dataset used in the validation was a small fraction of the whole training dataset.

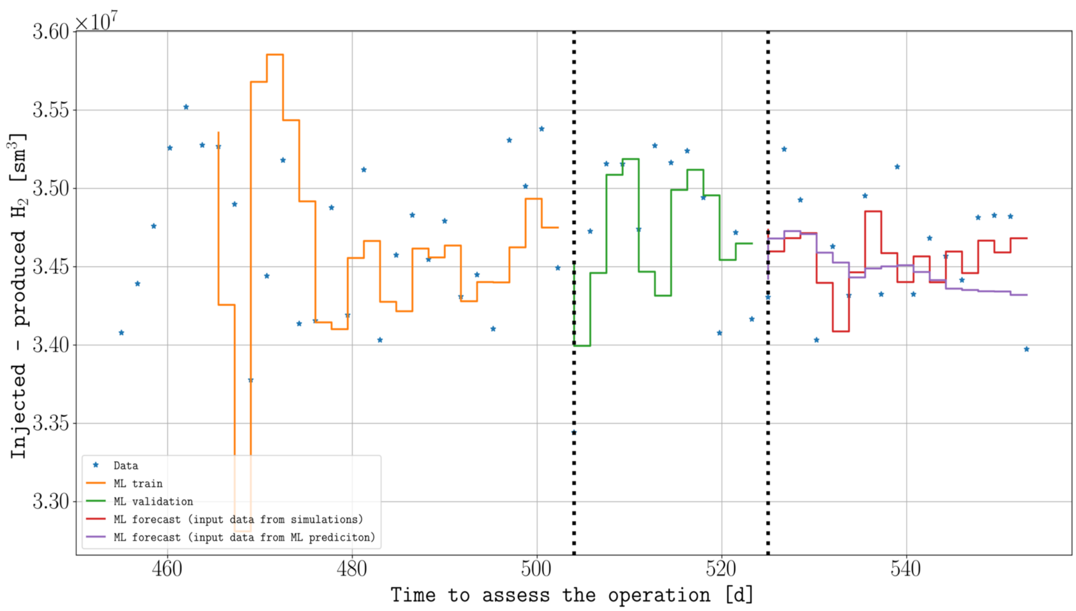
hydrogen volume, the prediction output is reliable with the model fitting between data points for any randomly selected time step. The three trends in the data (blue stars) arise from the scheduling operations which could be injection, storage, or production, with the injection and production cycles resulting highest and lowest residual hydrogen volume respectively. Comparing the computational time of 6 minutes per simulation run against 73 seconds for training a neural network able to make various predictions including the residual hydrogen volume validates the potential of ML in the industry. With the computational cost reduction value of 6773.5% in our study, it implies that the methodology has the potential to be applied to more complex full-field scale scenarios and achieve reliable outputs. It also implies that in case more field hydrogen injection, storage, and production data sets are provided, with proper fluid and grid model characterization the degree of uncertainty during UHS operations can be minimized. With more precise, fit-for-purpose, and optimized models, the uncertainty level can even be reduced further and yet make the industry more economically feasible. The findings in this study so far highlight the potential machine-learned models can achieve in the growing hydrogen industry. It adds to other geoscientific studies already conducted with this technology. For example [34] used three deep learning models to predict contact angle values in a multi-fluid rock system and compared their results against experimental and simulated datasets whereas [32] applied the same approach in making hydrogen-brine IFT value predictions amongst others.

3.3. Base case 2

The machine learning tasks on base case 2 were obtained on a dataset simulated on a one-layer reservoir model having a horizontal well grazing the top surface shown infigure 3. The same structure of sequential Keras NN model as in target scenario 1 is used. However, several hyper-parameters had to be tuned for the model to capture the data trend. The main difference was in the number of hidden layers, which was set to two. table 3 summarises and compares the hyper-parameters used in this scenario against those from base case 1. The same model training, validation, testing, and deployment approach is used. The results displayed in figure 13 and figure 14 are obtained from a simulation dataset that incorporated less fluid flow physics in the reservoir model. The only physics contributing to the amount of residual hydrogen saturation was dissolution and capillary pressure. The ML model precisely matched the injection data set as seen in figure 13. However, when it was deployed to predict on a production dataset after training and validation rounds, it performed



(A) Model fitting on the production dataset



(B) Model fitting on the residual hydrogen volume at ultimate production

FIGURE 12. Predictions on a more physical dataset described in target scenario 1. To fit the data, the depth of the neural network had to be increased to 6 layers and this also substantially increased the number of neurons in the network. In both cases, the model well fitted the dataset during training and validation steps as seen with orange and green curves. Subsequent predictions (red and purple curves) were also with acceptable ranges. However, a slight mismatch was observed on both predictions for both production (a) and residual hydrogen volume forecasts (b). The mismatch was associated with the physical complexity in the dataset.

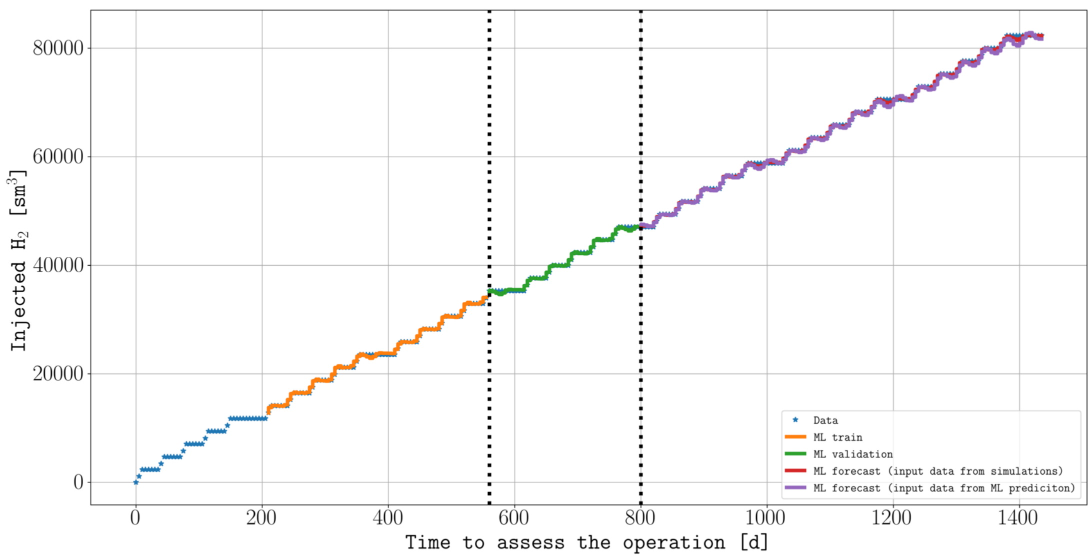
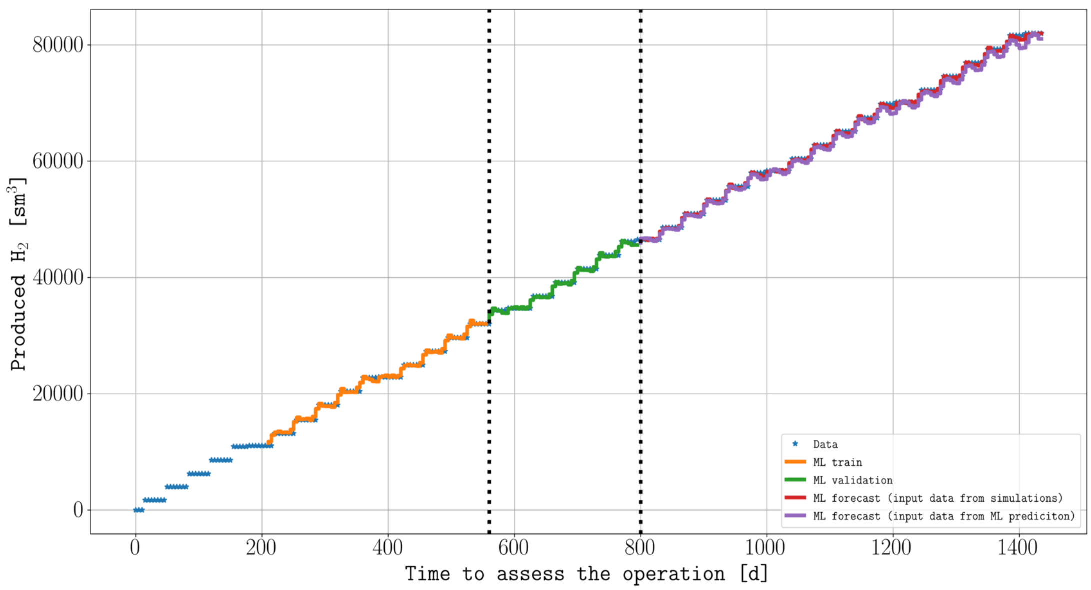


FIGURE 13. Model fitting on an injection dataset described in base case 2. The injection data set was used as a pretesting step to confirm the efficiency of the optimization round. The model was expected to over-fit the data as it did.

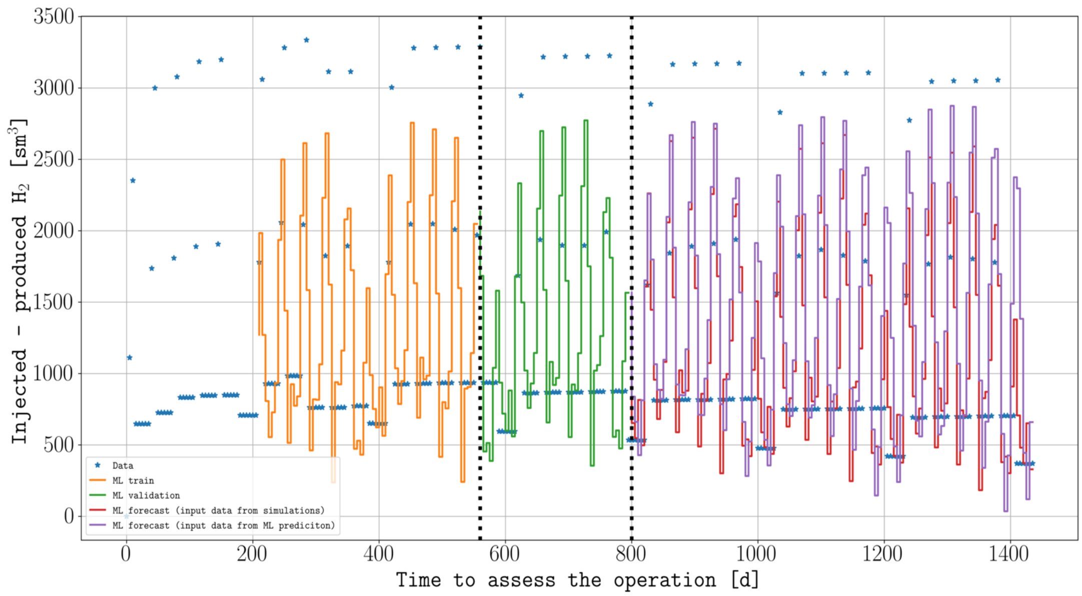
relatively well without overfitting the data set as seen in figure 14a. The trends of the residual hydrogen saturation were dependent on whether it was evaluated after injection, production, or storage rounds. With the values remaining relatively highest after injection, lowest after production, and relatively constant during storage. The three trends can be deduced from the three data clusters (the blue stars) in figure 14b. The whole training dataset was simulated on seven seasonal cyclic cycles that can be seen as abrupt breaks in trends in figure 14b. When the model was trained and deployed on such a dataset, capturing the cycles and seasonal trends in the dataset required a larger look back time stepping of 42 in the used windowed sequential Keras model. The batch size during model training also had to be increased to 21. All other model hyper-parameters were used in a default mode otherwise were global to all the training rounds described before. The trend in the data set was then captured by the model without overfitting the dataset. When the NN was called to predict new data sets, the model still performed relatively well and followed the same trends learned from training and validation rounds, shown in red and purple curves in figure 14b. To asses model performance during training and validation steps, the results of the loss function valuesMSE for each epoch are shown in figure 16a. In both steps, the MSE drastically drops in the first 200 epochs and stays relatively uniform within an acceptable range. With the validation endpoint MSE value dropping close to zero to about 5.1E-5, it implies the model does neither overfit nor underfit the data and can therefore be applied to any new data set with confidence. However, it is important to note that there are no two like reservoirs. Therefore, the data exposed to this learned model has got to be from this reservoir otherwise a new model has to be trained for any other screened reservoir model for underground/subsurface hydrogen storage (UHS) operations.

3.4. **Target scenario 2**

With more physics included in the reservoir model to have a better understanding of how much hydrogen can be recovered during cyclic injection, a new simulated dataset was used to train the same NN model. The results in figure 15 stem from the mentioned new dataset. The model is first trained and deployed on the production



(a) Prediction on the production dataset from base case 2



(b) Prediction on the residual hydrogen volume from base case 2

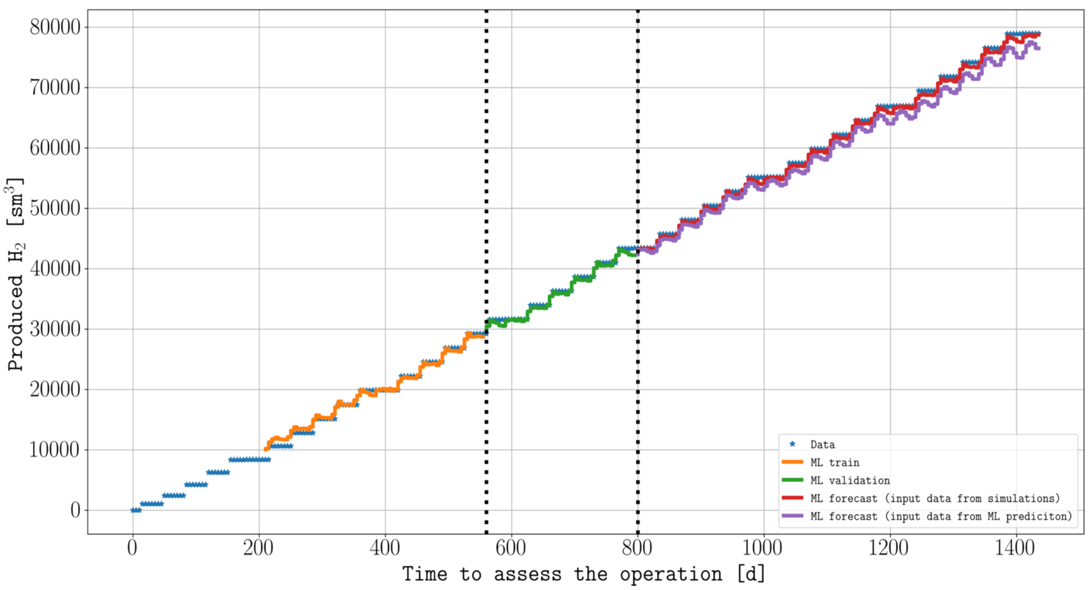
FIGURE 14. Model fitting and predictions on the production datasets were good as seen by all the curve fitting in (a). However, the training, validation and predictions on the residual hydrogen volume (b) was also good but mostly guided by the location of data values within the dataset. With lower values (1000-2000 sm^3) dominating influencing the model fitting process. This could potentially illustrate that its better to predict on the production dataset with a narrow range value.

dataset. However, to capture the data trends, a holistic hyper-parameter optimization had to be conducted first. The ultimate optimal model still has two hidden layers with 28 and 6 neurons connecting the second and third to the input and output layers respectively. The optimal batch size is 42 and the maximum number of iterations during optimisation is increased from 10 to 20. The rest of the key hyper-parameters are summarised in table 3. The newly updated NN model matches the data trend including the seven-season breaks during the training and validation rounds. When the model is called to predict on new data, the match from the new simulation data is better than that from the ML generated as can be seen by the red and purple curves in figure 15a respectively. As discussed above for the less physics dataset, the residual hydrogen volume is dependent on the operations history. Therefore, for the NN model to capture the inherent trend information contained in the dataset, a new random search optimization round has to be conducted. Fortunately, the resultant optimal model has the same hyper-parameters as for the production dataset. The same model is therefore applied to predict the residual volumes. The training and validation rounds matched the trends. However, the predictions on the simulated and ML newly generated data sets are slightly different as seen by the red and purple curves in figure 15b respectively. The residual volumes predicted on a simulated dataset are mostly lower than those on ML generated. Nonetheless, the two trends interpolated and ranged between the actual data points (blue stars). The model assessment parameters during training and validation rounds, figure 16b, reveals a relatively similar model performance as in base case 2. The loss functions drastically dropped close to zero in the first 200 epochs and remained at that value for the rest of the epochs with an endpoint value of 5.8E-5. The trends of the loss function are identical both during training and validation and can be deducted from the overlapping orange and blue curves in figure 16b. All other key information in addition to comparing simulation times against model training times is summarised in table 3 and table 4. The zeroing out of the loss function during training and validation of the machine-learned NN model learned on a multi-physics production and injection dataset gives confidence in the learned model. This can be related to the good predictions on the new dataset. The most important fluid flow parameter relevant for underground/subsurface hydrogen storage is relative permeability hysteresis resulting from contact angle changes due to continuous loading and unloading of the reservoir system as was highlighted by [26,45]. The cyclic behavior additionally exposes the system to drastic thermodynamic changes that would in a way accelerate the hysteric behavior at the rock-fluid interface. The complex injection, production, and storage schedule upon which the data was simulated, and the consideration of relative permeability hysteresis were the main parameters that had to be explored in this work. With the machine-learned NN model performing well on such a dataset, the methodology can then be utilized in reducing the uncertainty arising from short and long term UHS operations.

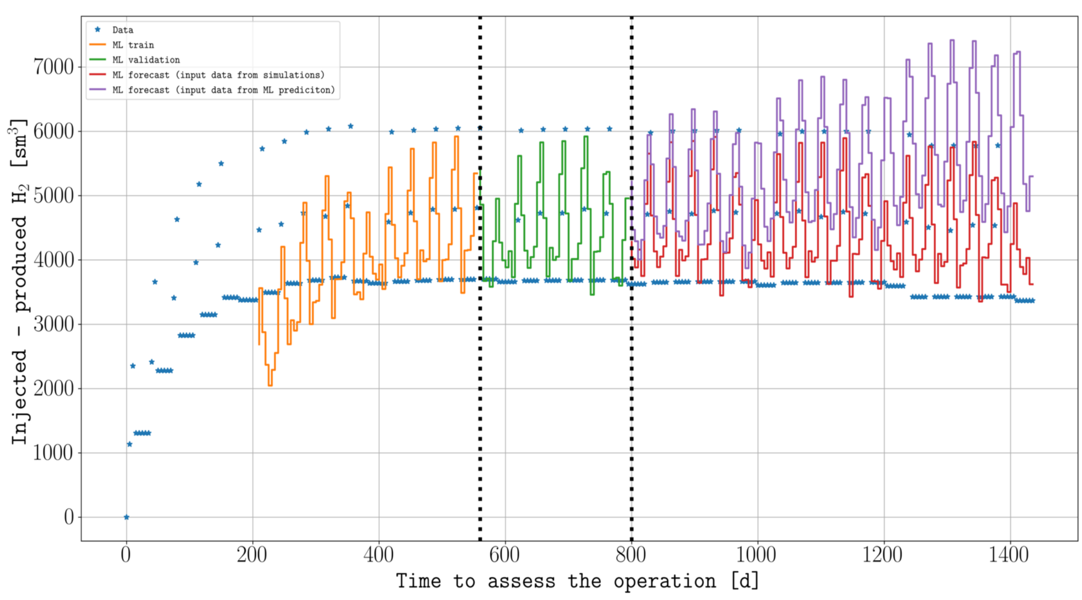
4. Conclusion and outlook

It has been shown in this work that machine-learned models can be used and play a crucial role in optimising underground/subsurface hydrogen storage (UHS) operations. The focus of the study was on the efficiency of subsurface operations where the models were tasked to show whether they could play a role that traditional simulators have historically performed. Multi-physics scenarios were simulated on published reservoir models upon which synthetic simulation data sets were generated. Comparisons between the simulation runs and machine-learned models were conducted. Based on our observations the following conclusions can be made from the study:

- A time series windowed keras model has been trained and used to make production forecasts within acceptable accuracy levels. The loss function values (the mean squared error (MSE) and mean absolute error (MAE)) support this conclusion.
- The same model successfully predicted the residual hydrogen volumes when called upon. The predictions were in an acceptable accuracy window using the same performance metrics.
- Predicting on the injection volumes seemed unnecessary because any neural network (NN) model used no matter its structure was able to make perfect predictions. In other words, the models perfectly matched the data sets. This was expected as the injected amount injected into the reservoir is normally well known.



(a)



(b)

FIGURE 15. Model fitting on a multi-physics production and residual hydrogen volume data sets. The production data trends were well captured by the learned model as can be seen in (a). Predictions on the new production dataset (red) and that machine-learned model generated (purple) were also good with the difference within an acceptable range. The fitting on the residual volume was good as shown in (b) even though a slight mismatch existed between the predictions (red and purple lines). The fitting seemed to have been guided by the regions in the data with high cluster concentrations (3000-4000 sm^3) (the blue stars).

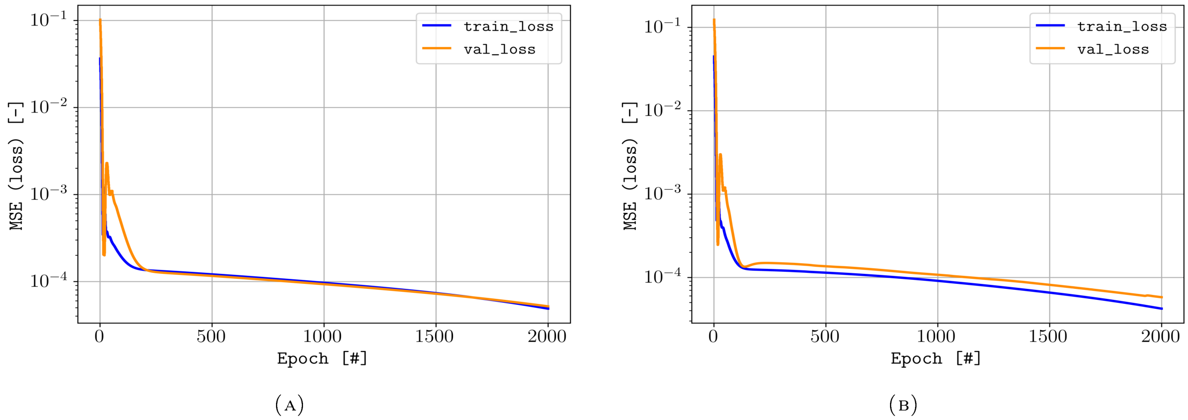


FIGURE 16. The loss function evaluations during the training and validation steps for base case 2 (a) and the multi-physics target scenario 2 (b). In both case, the loss monotonically decreased with the validation loss always higher than the training loss.

- Computational costs in terms of run time from the machine-learned predictions were far much smaller in comparison with either a single or a collection of simulation runs. A multi-layer multi-physics scenario had 4 orders of magnitude difference. That was equivalent to 6773.5% percentage difference.
- Predictions on the dataset generated on a complex UHS schedule using a one-layer reservoir model produced unique trends in the production data-sets. These trends were well captured by the learned neural network (NN) model and substantially reduced the computational cost for a 7-season scenario for in four years by 731.2%.

The target physics in the study focused on hydrogen loss in the reservoir upon production. Mainly its dissolution in the water phase and relative permeability hysteresis due to cyclic operations. It suggests that in the future additional parameters need to be considered in the reservoir model to capture a more multi-physics loss scenario. Such parameters include H_2 loss due to microbial activity, geochemical reactions, diffusion into the cap rock among others. The effect of reservoir geometry could also be considered as a contributing hydrogen (H_2) loss parameter. This study also considered the economic limit value of 95% but a more realistic limit could be used that matches the prevailing H_2 market needs. Future investigations will also consider larger models and more sophisticated architectures (such as LSTM, Deep operator networks among others).

Acknowledgement

The authors acknowledge funding from the Centre of Sustainable Subsurface Resources (CSSR), grant nr. 331841, supported by the Research Council of Norway, research partners NORCE Research AS and the University of Bergen, and user partners Equinor ASA, Harbour Energy, Sumitomo Corporation, Earth Science Analytics, GCE Ocean Technology, SLB Scandinavia in addition to the University of Stavanger and Petroleum Research School of Norway (NFiP).

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. {TensorFlow}: a system for {Large-Scale} machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pages 265–283, 2016.

[2] Hooman Abdollahi. A novel hybrid model for forecasting crude oil price based on time series decomposition. *Applied energy*, 267:115035, 2020.

[3] Farhad Abedi, Mahdi Moosavi, Abbas Bahroudi, and Alireza Moazenian. Effect of solid impurity on creep behavior of salt rocks of the hormoz formation. *International Journal of Mining and Geo-Engineering*, 54(2):161–166, 2020.

[4] Aliyuda Ali. Data-driven based machine learning models for predicting the deliverability of underground natural gas storage in salt caverns. *Energy*, 229:120648, 2021.

[5] Aliyuda Ali and Lingzhong Guo. Adaptive neuro-fuzzy approach for prediction of dewpoint pressure for gas condensate reservoirs. *Petroleum Science and Technology*, 38(9):673–681, 2020.

[6] Menad Nait Amar. Towards improved genetic programming based-correlations for predicting the interfacial tension of the systems pure/impure co2-brine. *Journal of the Taiwan Institute of Chemical Engineers*, 127:186–196, 2021.

[7] Menad Nait Amar, Ashkan Jahanbani Ghahfarokhi, and Noureddine Zeraibi. Predicting thermal conductivity of carbon dioxide using group of data-driven models. *Journal of the Taiwan Institute of Chemical Engineers*, 113:165–177, 2020.

[8] Susan Anyosa, Scott Bunting, Jo Eidsvik, Anouar Romdhane, and Per Bergmo. Assessing the value of seismic monitoring of co2 storage using simulations and statistical analysis. *International Journal of Greenhouse Gas Control*, 105:103219, 2021.

[9] Peter Bastian, Markus Blatt, Andreas Dedner, Christian Engwer, Robert Klöfkorn, Mario Ohlberger, and Oliver Sander. A generic grid interface for parallel and adaptive scientific computing. part i: abstract framework. *Computing*, 82:103–119, 2008.

[10] D. Baxendale and et al. Opm flow documentation manual, 2023. `https://opm-project.org/?page_id=955`.

[11] Ian H. Bell, Jorrit Wronski, Sylvain Quoilin, and Vincent Lemort. Pure and pseudo-pure fluid thermophysical property evaluation and the open-source thermophysical property library coolprop. *Industrial & Engineering Chemistry Research*, 53(6):2498–2508, 2014.

[12] Erik G Boman, Ümit V Çatalyürek, Cédric Chevalier, and Karen D Devine. The zoltan and isorropia parallel toolkits for combinatorial scientific computing: Partitioning, ordering and coloring. *Scientific Programming*, 20(2):129–150, 2012.

[13] Francois Chollet et al. Keras, 2015.

[14] Katriona Edlmann, Sofi Hinchliffe, Niklas Heinemann, Gareth Johnson, Jonathan Ennis-King, and CI McDermott. Cyclic co2–h2o injection and residual trapping: implications for co2 injection efficiency and storage security. *International Journal of Greenhouse Gas Control*, 80:1–9, 2019.

[15] Bernd Flemisch, Melanie Darcis, K Erbertseder, B Faigle, A Lauser, Klaus Mosthaf, S Müthing, Philipp Nuske, A Tatomir, M Wolff, et al. Dumux: Dune for multi-{phase, component, scale, physics,...} flow and transport in porous media. *Advances in Water Resources*, 34(9):1102–1112, 2011.

[16] Saeed Harati, Sina Rezaei Gomari, Firdovsi Gasanzade, Sebastian Bauer, Tannaz Pak, and Caroline Orr. Underground hydrogen storage to balance seasonal variations in energy demand: Impact of well configuration on storage performance in deep saline aquifers. *International Journal of Hydrogen Energy*, 2023.

[17] Jason E Heath, Sean A McKenna, Thomas A Dewers, Jesse D Roach, and Peter H Kobos. Multiwell co2 injectivity: impact of boundary conditions and brine extraction on geologic co2 storage efficiency and pressure buildup. *Environmental science & technology*, 48(2):1067–1074, 2014.

[18] Niklas Heinemann, Juan Alcalde, Johannes M Miocic, Suzanne JT Hangx, Jens Kallmeyer, Christian Ostertag-Henning, Aliakbar Hassanpouryouzband, Eike M Thaysen, Gion J Strobel, Cornelia Schmidt-Hattenberger, et al. Enabling large-scale hydrogen storage in porous media–the scientific challenges. *Energy & Environmental Science*, 14(2):853–864, 2021.

[19] Yinghua Jiang, Lixia Kang, and Yongzhong Liu. Optimal configuration of battery energy storage system with multiple types of batteries based on supply-demand characteristics. *Energy*, 206:118093, 2020.

[20] Nikhil Ketkar and Nikhil Ketkar. Introduction to keras. *Deep learning with python: a hands-on introduction*, pages 97–111, 2017.

[21] Chérifa Kara Mostefa Khelil, Badia Amrouche, Abou soufiane Benyoucef, Kamel Kara, and Aissa Chouder. New intelligent fault diagnosis (ifd) approach for grid-connected photovoltaic systems. *Energy*, 211:118591, 2020.

[22] J.E. Killough. Reservoir simulation with history-dependent saturation functions. *Society of Petroleum Engineers Journal*, 16(01):37–48, feb 1976.

[23] CA Kossack. Comparison of reservoir simulation hysteresis options. In *SPE Annual Technical Conference and Exhibition?*, pages SPE–63147. SPE, 2000.

[24] D. Landa-Marbán and P. M. von Schultzendorff. pyopmnearwell: A framework to simulate near well dynamics using opm flow, v2023.10. `https://doi.org/10.5281/zenodo.10266791`.

[25] Dedong Li, Christof Beyer, and Sebastian Bauer. A unified phase equilibrium model for hydrogen solubility and solution density. *International Journal of Hydrogen Energy*, 43(1):512–529, January 2018.

[26] Maksim Lysyy, Tore Føyen, Else Birkeland Johannesen, Martin Fernø, and Geir Ersland. Hydrogen relative permeability hysteresis in underground storage. *Geophysical Research Letters*, 49(17):e2022GL100364, 2022.

[27] Mohammad Madani, Peyman Abbasi, Alireza Baghban, Ghasem Zargar, and Pezhman Abbasi. Modeling of co2-brine interfacial tension: Application to enhanced oil recovery. *Petroleum Science and Technology*, 35(23):2179–2186, 2017.

[28] Hossein Mehrjoo, Mohsen Riazi, Menad Nait Amar, and Abdolhossein Hemmati-Sarapardeh. Modeling interfacial tension of methane-brine systems at high pressure and high salinity conditions. *Journal of the Taiwan Institute of Chemical Engineers*, 114:125–141, 2020.

[29] Nasiru Salahu Muhammed, Bashirul Haq, Dhafer Al Shehri, Amir Al-Ahmed, Mohammed Mizanur Rahman, and Ehsan Zaman. A review on underground hydrogen storage: Insight into geological sites, influencing factors and future outlook. *Energy Reports*, 8:461–499, 2022.

[30] Cuthbert Shang Wui Ng, Hakim Djema, Menad Nait Amar, and Ashkan Jahanbani Ghahfarokhi. Modeling interfacial tension of the hydrogen-brine system using robust machine learning techniques: Implication for underground hydrogen storage. *international journal of hydrogen energy*, 47(93):39595–39605, 2022.

[31] Esuru Rita Okoroafor, Luiz Sampaio, Firdovsi Gasanzade, Yulman Perez Claro, Jimin D. Zhou, Sarah D. Saltzer, Sebastian Bauer, and Anthony R. Kovscek. Intercomparison of numerical simulation models for hydrogen storage in porous media using different codes. *Energy Conversion and Management*, 292:117409, 2023.

[32] Sina Omrani, Mehdi Ghasemi, Mrityunjay Singh, Saeed Mahmoodpour, Tianhang Zhou, Masoud Babaei, and Vahid Niasar. Interfacial tension–temperature–pressure–salinity relationship for the hydrogen–brine system under reservoir conditions: Integration of molecular dynamics and machine learning. *Langmuir*, 39(36):12680–12691, 2023.

[33] Alessandra Perna, Mariagiovanna Minutillo, and Elio Jannelli. Hydrogen from intermittent renewable energy sources as gasification medium in integrated waste gasification combined cycle power plants: A performance comparison. *Energy*, 94:457–465, 2016.

[34] Arash Rabbani, Chenhao Sun, Masoud Babaei, Vahid J Niasar, Ryan T Armstrong, and Peyman Mostaghimi. Deepangle: Fast calculation of contact angles in tomography images using deep learning. *Geoenergy Science and Engineering*, 227:211807, 2023.

[35] Atgeirr Flø Rasmussen, Tor Harald Sandve, Kai Bao, Andreas Lauser, Joakim Hove, Bård Skaflestad, Robert Klöfkorn, Markus Blatt, Alf Birger Rustad, Ove Sævareid, Knut-Andreas Lie, and Andreas Thune. The open porous media flow reservoir simulator. *Computers & Mathematics with Applications*, 81:159–185, 2021. Development and Application of Open-source Software for Problems with Numerical PDEs.

[36] FN Schneider and WW Owens. Relative permeability studies of gas-water flow following solvent injection in carbonate rocks. *Society of Petroleum Engineers Journal*, 16(01):23–30, 1976.

[37] P. Von Schultzendorff, T.H. Sandve, B. Kane, D. Landa-Marbán, J.W. Both, and J.M. Nordbotten. A machine-learned near-well model in opm flow. 2024(1):1–23, 2024.

[38] Mehran Sohrabi, Ali Danesh, Dabir H Tehrani, and Mahmoud Jamiolahmady. Microscopic mechanisms of oil recovery by near-miscible gas injection. *Transport in Porous Media*, 72:351–367, 2008.

[39] Youngsoo Song, Wonmo Sung, Youngho Jang, and Woodong Jung. Application of an artificial neural network in predicting the effectiveness of trapping mechanisms on co2 sequestration in saline aquifers. *International Journal of Greenhouse Gas Control*, 98:103042, 2020.

[40] Reza Tahmasebifar, Mohsen Parsa Moghaddam, Mohammad Kazem Sheikh-El-Eslami, and Reza Kheirollahi. A new hybrid model for point and probabilistic forecasting of wind power. *Energy*, 211:119016, 2020.

[41] Evan Schankee Um, David Alumbaugh, Youzuo Lin, and Shihang Feng. Real-time deep-learning inversion of seismic full waveform data for co2 saturation and uncertainty in geological carbon storage monitoring. *Geophysical Prospecting*, 2022.

[42] Yue Wang, Ridoy Das, Ghanim Putrus, and Richard Kotter. Economic evaluation of photovoltaic and energy storage technologies for future domestic energy systems–a case study of the uk. *Energy*, 203:117826, 2020.

[43] Seongcheol Woo, Junyoung Park, Jinkyoo Park, and Lance Manuel. Wind field-based short-term turbine response forecasting by stacked dilated convolutional lstms. *IEEE Transactions on Sustainable Energy*, 11(4):2294–2304, 2019.

[44] Zhi Yuan, Weiqing Wang, Haiyun Wang, and Scott Mizzi. Combination of cuckoo search and wavelet neural network for midterm building energy forecast. *Energy*, 202:117728, 2020.

[45] Haiyang Zhang, Yihuai Zhang, Mohammed Al Kobaisi, Stefan Iglauer, and Muhammad Arif. Effect of cyclic hysteretic multiphase flow on underground hydrogen storage: A numerical investigation. *International Journal of Hydrogen Energy*, 2023.

[46] Jiyuan Zhang, Qihong Feng, Shuhua Wang, Xianmin Zhang, and Shoulei Wang. Estimation of co2–brine interfacial tension using an artificial neural network. *The Journal of Supercritical Fluids*, 107:31–37, 2016.