International Journal of Greenhouse Gas Control

Utilisation of Artificial Intelligence based Time-Series Prediction to validate Carbon Containment in Injection Well in Illinois Basin

--Manuscript Draft--

Manuscript Number:	JGGC-D-23-00588
Article Type:	Full Length Article
Keywords:	low-carbon, time-series, neural network, LSTM, carbon capture, injection pressure, monitoring, plume migration
Corresponding Author:	Munish Kumar Singapore University of Social Sciences SINGAPORE
First Author:	Munish Kumar
Order of Authors:	Munish Kumar
	Kannappan Swaminathan
Abstract:	Carbon Capture Utilisation and Storage (CCUS) involves capturing CO2 emissions and securely storing them in geological formations. CCUS is gaining significance in global efforts to meet ambitious climate goals. The storage of CO2 typically occurs through injection into saline aquifers, depleted oil and gas fields, or for Enhanced Oil Recovery (EOR). However, all methods require a deep understanding of subsurface geology and the ability to monitor CO2 behaviour during injection and storage. Pilot projects like the Illinois Basin - Decatur Project demonstrated the practical feasibility of storing CO2 underground. The injection spanned three years, during which nearly 999,215 tonnes of CO2 was stored. Monitoring was carried out through a pair of wells equipped with sensors to track pressure and temperature at various depths. This paper focuses on using time series injection data and monitoring information to predict changes in injection rates for the carbon capture well. We perform the prediction using Long Short-Term Memory (LSTM) neural networks (NN). These changes, represented as deltas (Δ) in injection rates between time t and time (t-1), are crucial indicators of carbon containment and migration within the well. By correlating these rate changes with other well parameters, this approach serves as a checkpoint against unwanted carbon migration or losses during the injection process. Machine learning methods are applied to forecast these injection rate deltas based on monitoring data, thereby validating the effectiveness of carbon containment during injection.
Suggested Reviewers:	Alessandro Romagnoli A.Romagnoli@ntu.edu.sg
	PETER FANTKE PEFAN@DTU.DK
	Kamaljit Singh K.Singh@hw.ac.uk
	Praveen Linga chepl@nus.edu.sg
	Ning YAN ning.yan@nus.edu.sg
	Karen Lythgoe karen.lythgoe@ed.ac.uk

Cover letter to the Editor

Manuscript title: Utilisation of Artificial Intelligence based Time-Series Prediction to validate Carbon Containment in Injection Well in Illinois Basin

Name of the Corresponding Author: Munish Kumar
Name(s) of all other authors: Kannappan Swaminathan
Type of Manuscript: Research Article

This manuscript is appropriate for *the International Journal of Greenhouse Gas Control (IJGGC)* as to this authors knowledge, there is a limited pool of information combining engineering, data science and artificial intelligence for time series prediction and forecasting of fields undergoing carbon capture and storage. We primarily think this is due to the lack of publicly available data on how CO_2 wells behave once injection begins. We are both practicing energy professionals and hope that our work can be utilized by other like-minded scientists as they attempt to predict injection well behavior and better identify anomalies to ensure CO_2 containment in the long run.

We have realized, through this work, that having a holistic understanding of both practical petroleum engineering and data science within the carbon capture and storage realm can either help to derisk opportunities, or at the very minimum, explain why a project may perform in a sub-optimal manner. This makes our work important and the journal the appropriate avenue to publish it.

The manuscript has been checked by a native English speaker with expertise in the field of energy. In this authors opinion, this work would appeal to both a popular audience and scientific audience.

The manuscript, or its contents in some other form, has not been published previously by the author and is not under consideration for publication in another journal at the time of submission. The manuscript does not have any supporting information and/or Review-Only Material.

Yours sincerely,

Dr Munish Kumar
October 03, 2023

Highlights

- CCUS is one method to deal with produced CO2.
- Prediction how the CO2 behaves in the subsurface once injected is difficult.
- We aim to use LSTM neural networks to predict injection deltas, which are changes that occur during the injection process.
- The input data source comprises measurements taken from sensors located a distance away in a monitoring well.
- Predicting this change can be used to provide a checkpoint against carbon plume migration and can determine if there are losses in the injection process.

Title: Utilisation of Artificial Intelligence based Time-Series Prediction to validate Carbon Containment in Injection Well in Illinois Basin

Authors: Munish Kumar[a] and Kannappan Swaminathan[b]

Affiliations: [a]Singapore University of Social Sciences, School of Business, Clementi Rd, #463号, Singapore 599494

[b]Insights Inc

Corresponding author. Email: munishkumar001@suss.edu.sg

Abstract:

Carbon Capture Utilisation and Storage (CCUS) involves capturing CO_2 emissions and securely storing them in geological formations. CCUS is gaining significance in global efforts to meet ambitious climate goals. The storage of CO2 typically occurs through injection into saline aquifers, depleted oil and gas fields, or for Enhanced Oil Recovery (EOR). However, all methods require a deep understanding of subsurface geology and the ability to monitor CO_2 behaviour during injection and storage.

Pilot projects like the Illinois Basin - Decatur Project demonstrated the practical feasibility of storing CO_2 underground. The injection spanned three years, during which nearly 999,215 tonnes of CO_2 was stored. Monitoring was carried out through a pair of wells equipped with sensors to track pressure and temperature at various depths.

This paper focuses on using time series injection data and monitoring information to predict changes in injection rates for the carbon capture well. We perform the prediction using Long Short-Term Memory (LSTM) neural networks (NN). These changes, represented as deltas (Δ) in injection rates between time t and time (t-1), are crucial indicators of carbon containment and migration within the well. By correlating these rate changes with other well parameters, this approach serves as a checkpoint against unwanted carbon migration or losses during the injection process. Machine learning methods are applied to forecast these

injection rate deltas based on monitoring data, thereby validating the effectiveness of carbon containment during injection.

One-Sentence Summary: Applying machine learning and predictive analytics via time series injection information and monitoring data on a carbon capture well to predict well injection rate deltas.

Keywords (minimum 6): low-carbon, time-series, neural network, LSTM, carbon capture, injection pressure, monitoring, plume migration

Introduction

Undoubtedly, human activities stand as a prominent cause behind the surge in greenhouse gas (GHG) emissions and the subsequent global warming. The escalation of carbon dioxide (CO_2) levels can primarily be attributed to swift industrialization and population expansion, which have only accelerated since the 1960s [1].

The burning of fossil fuels (coal, oil, and natural gas) has resulted in carbon dioxide (CO_2), methane (CH_4) and nitrous oxide (N_2O) generation. A report generated by the United States Environmental Protection Agency (EPA) evaluated the amount of CO_2 generated in 2020 to be ~3.11 million metric tons [2]. Failure to effectively manage such substantial quantities will result in irreversible environmental consequences.

Carbon Capture Utilisation and Storage (CCUS) is one promising method to deal with the produced CO_2 from anthropogenic sources. The idea is to capture CO_2 from an emission point source and subsequent sequester it via injection into a suitable geological formation, with the explicit aim to store the CO_2 safely, in a state of permanence. CCUS is now being viewed as a key technology that will assist us in reaching increasingly ambitious global anthropogenic climate change goals. Typically, CO_2 storage is carried out in one of three ways - via injection into virgin saline aquifers; into depleted oil and gas fields; or if used for Enhanced Oil Recovery (EOR) processes. These methods all have different project drivers, risks, and commercial implications. However, what these 3 methods have in common is the requirement that (a) there be a good understanding of the subsurface geological properties and (b) there be some ability to monitor and even predict CO_2 behaviour at the well scale, be it during the injection phase or during the shut-in phase.

Given the early stage of this technique, it is imperative to initiate pilot development and validate the technology. The Illinois Basin - Decatur Project is one such study which aimed to demonstrate the capacity, injectivity, and containment of carbon storage in the Mount Simon Sandstone, the main carbon storage resource in the Illinois Basin and the Midwest Region. The source of the injected CO_2 is from ethanol production at the Archer Daniels Midland company's plant. The CO_2 is compressed, dehydrated and injected

into the Mt. Simon Sandstone, which is primarily a saline aquifer approximately ~7,000 ft deep. Injection began in 2009 and continued for a 3-year period (Nov-2011 to Nov-2014). Cumulatively, ~999,215 tonnes of supercritical CO_2 have been injected and geologically stored. A pair of injection and verification wells, ~700ft apart, were drilled into the formation. The wells were equipped with downhole sensors to monitor pressure and temperature at various depths of interest.

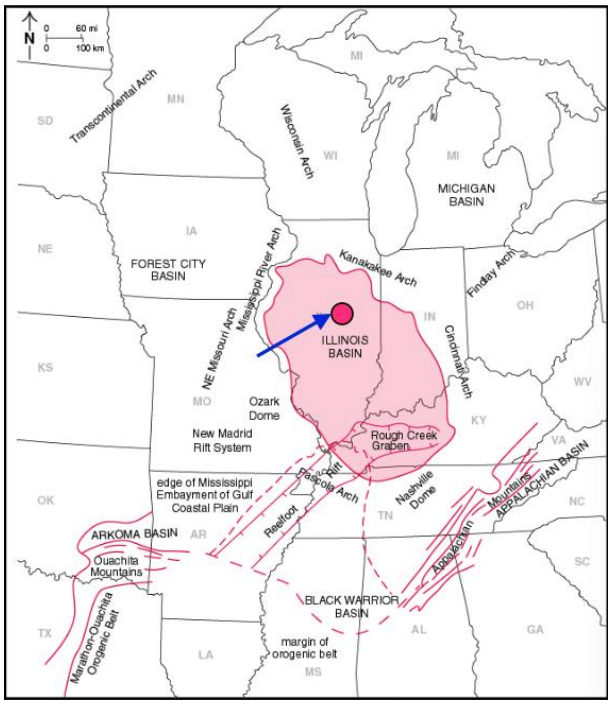


Figure 1: Location map of the Illinois Basin – Decatur Project (IBDP). Image taken from [3]

This paper aims to use time series injection information and monitoring data on a carbon capture well to predict carbon capture well injection rates deltas (Δ) which is the difference in the injection rate (IR) at time t and time (t-1) i.e.

$$\Delta = IR_t - IR_{t-1} \qquad \text{Equation 1}$$

Correlating the change in injection rate to the behaviour of other parameters in the well can be used to provide a checkpoint against carbon migration from the well or other losses during the process. Utilisation of

a machine learning (ML) method to predict injection rate deltas based on monitoring well data can be used to validate carbon containment throughout the injection of the well as well.

Literature Review

Various authors have tried numerous methods to forecast future trends based on past data. Work by De Gooijer and Hyndman [4], for instance, reviewed a series of time-series forecast models over a 25-year period, from 1985 to 2005. Their review highlighted various models being developed and applied in a myriad of scenarios related to finance, statistics and manufacturing, and included methods such as (a) exponential smoothing [5, 6], (b) Autoregressive Integrated Moving Average (ARIMA) [7], (c) seasonal models [8], (d) state space and structural models and the Kalman filter [9], (e) nonlinear models [10], (f) long-range dependence models, including the family of Autoregressive Fractionally Integrated Moving Average (ARFIMA) models [11], (g) Autoregressive Conditional Heteroskedasticity/Generalized Autoregressive Conditional Heteroskedasticity (ARCH/GARCH) models [12], and (h) count data forecasting [13].

Within reservoir engineering, the prediction of hydrocarbon and water rates from geological formations can be attributed as a time-series forecasting problem. Empirical solutions developed by Arps [14], referred to as decline curve analysis (DCA) technique is one of the earliest methods to address this problem. The method is based on a curve-fit principle, where one would attempt to fit either exponential, hyperbolic or harmonic curve to historical flow production rate as a function of time. Equation 2 shows the general form of the equation:

$$q(t) = \frac{q_i}{(1 + bD_i t)^{1/b}} \qquad \text{Equation 2}$$

where q_i is the initial rate (bbls/day), D_i is the initial decline rate (units) and b is the degree of curvature of the line. An exponential curve fit would have b = 0, a hyperbolic curve would have 0 < b < 1 and a harmonic curve would have b = 1. The fitted curve is then used to predict future production rates and cumulative

production. This method was originally designed to work with high porosity-permeability reservoirs and tends to overestimate hydrocarbon recovery from unconventional (low permeability) reservoirs. Thus, various authors have tried to expand on this work [15, 16, 17, 18], and are mostly variations of the initial DCA method developed by Arps. For an effective DCA forecast, domain and field knowledge is key, but inherently the process is one of trial and error, and thus it is not uncommon for DCA results to have low-best-high estimates.

With the advent of big data, fast computing and cheap memory, applying a machine learning (ML) and artificial intelligence (AI) solution for time-series forecasting seems a natural evolution. ML solutions were first introduced to the petroleum industry in the early 2000s. Applications of ML and AI include addressing prediction of reservoir parameters [19], history matching, of oil, gas and water production forecasting (flow rate prediction), pattern recognition in well logs and well tests analysis, production enhancement and prediction of failures, among others [20, 21, 22].

In the field of CCUS and injector production performance, there is little to no available data related to time series and forecasting. This can be attributed to the early stages of development of both domains. Literature appears to be mostly concentrated in predicting carbon emissions [23], leakage [24], CO2 absorption and adsorption [25], property prediction and process simulation [26], simulation of transportation, and geological behaviours as it relates to uncertainty analysis, sequestration, utilisation and EOR processes [23, 24, 25] Work by Iskander et al employs Long Short-Term Memory (LSTM) networks to forecast oil, water and CO_2 production at future infill well locations, for both single phase and 3-phase fluid models. Data was in the form of a synthetic PUNQ-S3 reservoir model, combined with real-world observations from 8 production wells, which recorded daily production volumes over a decade from 2004-2014 [26]. Injection data from CO_2 wells was not a direct input in the deep learning model, although it did play an indirect role in the oil and gas production data being recorded, as it swept the residual hydrocarbons and therefore resulted in an uplift in production rates.

We aim to develop on the work of Iskander and others by utilising ML and AI methods, and in particular LSTM, but focusing on the prediction of CCUS injection well performance, using the open-source information from the IBDP (Illinois Basin Decatur Project). We will demonstrate how our developed LSTM model shows a correlation between the change in injection rate to the behaviour of other dynamic parameters [26].

While the primary purpose of the model is as a checkpoint against carbon migration, either at the well location or from other losses during the injection process, we view the model as another means for engineers to perform scenario based de-risking of exploration plays, via modelling variation in well and storage parameters to validate CO_2 containment. The model will also aid in the understanding of the injection process and potentially can be used to "right size" well operations and optimise costs.

Scope and Methods

Datasets

During this three-year period, a substantial amount of data was collected from both an injection and monitoring well, 700 ft apart. The injection well was drilled to a total depth (TD) of ~7051 ft, and was drilled with a 26" bit to 355 ft, and cased with a 20" casing to surface. A 17 ½" hole size then followed to a TD of 5339 ft, and an intermediate casing string 13 3/8" in diameter was set. The reservoir section was drilled in a 12 ½" hole size to ~ 7056 ft, and completed with a 9 5/8" production casing and 4 ½" tubing. The perforations were made at the base (i.e. above the pre-Cambrian) Mt. St Simon Sandstone, which was a relatively thick reservoir of ~1620 ft. A total of 3 geophones were set at 4925 ft, 5743 ft and 6137 ft along with a pressure / temperature gauge mandrel at 6325 ft. The monitor well was drilled to a total TD of 7272 ft; it had a surface casing (13 3/8" to 377 ft, followed by intermediate casing of 9 5/8" to 5322 ft and 5 ½" casing across the Mt. St Simon Sandstone, which contained a 3 component geophone array) [3].

A. Data Collection and Preparation

A total of 34 parameters were measured from the injection and verification well. The parameters measured comprised of both surface and downhole measurements which were acquired at five second intervals over three full years. The sheer volume of data required that the time scale be downscaled to hourly intervals, taking the parameter average over the hour. 27,665 hours of data (training data) was used to build a suitable model. 67% of the "training" data set was used to train the model. The remaining 33% was used as a "validation" dataset. The model built off this data was finally used to predict 201 hours of Δ into the future ("hold out" data). Given in Table 1 is the descriptive statistics of the provided data. We note that there are a series of "null values" and non-numeric numbers, missing rows and note that for some of the input data measurements, there are significant outliers.

Table 1: Data Statistics

Measurement	Non-Zero Value	Mean Value	Standard Deviation	Minimum Value	25th Percentile	Median Value	75th Percentile	Maximum Value
Avg_PLT_CO2VentRate_TPH	27398.0	2.1	133.2	0.0	0.0	0.1	0.2	18333.2
Avg_CCS1_WHCO2InjPs_psi	27270.0	1239.9	817.7	0.0	1235.5	1338.9	1361.0	39032.4
Avg_CCS1_WHCO2InjTp_F	27398.0	89.8	48.3	0.0	93.0	96.3	96.9	2879.4
Avg_CCS1_ANPs_psi	27304.0	560.9	445.9	0.0	523.5	564.9	604.8	24105.6
Avg_CCS1_DH6325Ps_psi	27398.0	3244.2	173.5	0.0	3233.0	3286.1	3324.7	3515.9
Avg_CCS1_DH6325Tp_F	27398.0	127.7	7.2	0.0	127.2	130.1	131.1	135.7
Avg_VW1_WBTbgPs_psi	26127.0	1801.8	999.4	0.0	2173.5	2322.4	2379.8	4954.7
Avg_VW1_WBTbgTp_F	26061.0	80.8	44.3	0.0	103.4	104.2	105.0	120.1
Avg_VW1_ANPs_psi	23487.0	525.0	3988.7	0.0	0.5	4.7	16.9	31993.5
Avg_VW1_Z11D4917Ps_psi	26688.0	1597.5	873.3	0.0	2070.3	2073.4	2074.1	2378.0
Avg_VW1_Z11D4917Tp_F	26709.0	81.7	44.2	0.0	103.7	105.2	106.5	108.6
Avg_VW1_Z10D5001Ps_psi	26688.0	1627.9	889.9	0.0	2106.9	2112.4	2116.6	2420.6
Avg_VW1_Z10D5001Tp_F	26709.0	81.2	44.0	0.0	101.8	104.7	105.0	110.9
Avg_VW1_Z09D5653Ps_psi	26688.0	1961.1	1071.9	0.0	2534.0	2547.5	2551.1	2785.4
Avg_VW1_Z09D5653Tp_F	26709.0	87.4	47.3	0.0	111.5	112.8	113.5	114.9

Avg_VW1_Z08D5840Ps_psi	26189.0	1604.8	1289.3	0.0	0.0	2627.3	2637.4	4446.2
Avg_VW1_Z08D5840Tp_F	25878.0	69.0	56.4	0.0	0.0	114.0	115.0	353.2
Avg_VW1_Z07D6416Ps_psi	25985.0	2199.2	1273.0	0.0	0.0	2911.6	2925.3	3195.0
Avg_VW1_Z07D6416Tp_F	25985.0	88.8	50.8	0.0	116.5	116.9	118.6	145.1
Avg_VW1_Z06D6632Ps_psi	25500.0	2315.6	1308.6	0.0	3012.2	3026.1	3031.3	3380.4
Avg_VW1_Z06D6632Tp_F	25500.0	90.9	50.8	0.0	116.6	118.6	119.4	124.3
Avg_VW1_Z05D6720Ps_psi	23955.0	2106.0	1447.2	0.0	0.0	3069.6	3073.6	3365.9
Avg_VW1_Z05D6720Tp_F	23955.0	82.3	55.0	0.0	0.0	118.5	119.4	122.1
Avg_VW1_Z04D6837Ps_psi	26600.0	2347.5	1369.0	0.0	0.0	3148.0	3153.5	3331.9
Avg_VW1_Z04D6837Tp_F	26600.0	91.0	51.6	0.0	118.8	119.5	119.9	125.8
Avg_VW1_Z03D6945Ps_psi	24361.0	2350.9	1495.7	0.0	0.0	3299.9	3320.6	3457.9
Avg_VW1_Z03D6945Tp_F	25932.0	182.3	368.5	0.0	0.0	121.3	122.8	1602.9
Avg_VW1_Z02D6982Ps_psi	26423.0	2456.6	1459.1	0.0	0.0	3316.2	3332.0	3499.6
Avg_VW1_Z02D6982Tp_F	26423.0	91.3	52.7	0.0	32.0	121.3	122.0	124.3
Avg_VW1_Z01D7061Ps_psi	25307.0	2300.2	1521.4	0.0	0.0	3318.2	3327.3	3445.1
Avg_VW1_Z01D7061Tp_F	25108.0	85.9	55.4	0.0	0.0	121.4	122.6	133.9
Avg_VW1_Z0910D5482Ps_psi	26709.0	1855.6	1017.6	0.0	2353.3	2374.9	2416.1	2758.3
Avg_VW1_Z0910D5482Tp_F	26709.0	86.3	46.7	0.0	110.5	111.5	112.0	113.7
inj_diff	27397.0	0.0	82.7	-11021.1	-0.1	0.0	0.1	7033.5

B. Cleaning Data

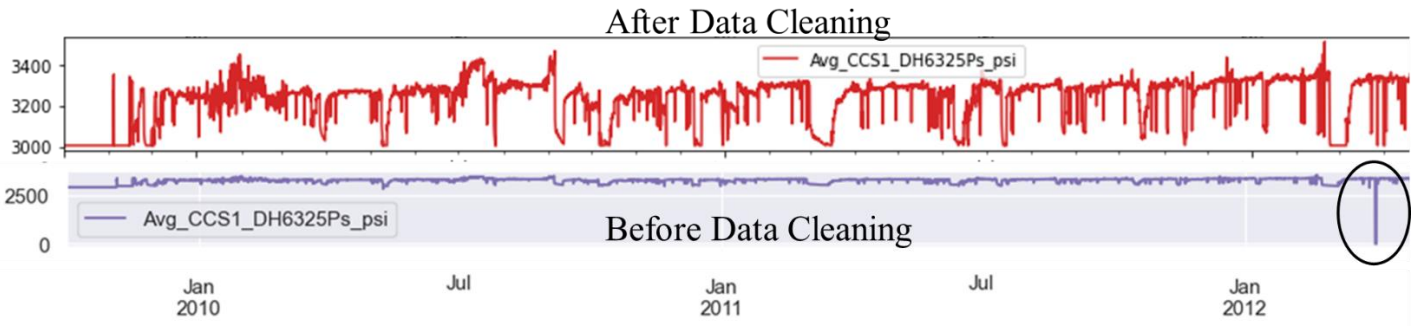
We reviewed the attributes for the data, and were especially concerned about discontinues, non-numerical data or data that were clearly outliers (e.g. ~25,000 psi WHP at injection well or 0 psi downhole gauge pressure). We utilised 3 methods to clean the data, in order of operation (i) we firstly performed a computational fill of all the missing and null values with a "forward fill' operation, with the last valid observation being propagated forward. We do this on the assumption that missing values retain the properties of the previous cell i.e. there has been no change in the data between time *t* and time (*t-1*), (ii) a Z-score method where

$$Z = \frac{x_i - \tilde{x}}{MAD}$$ Equation 3

177

178 x_i is a single data value, $\tilde{x}$ is the median of the data set and MAD is the median absolute deviation of the
179 dataset and (iii) a visual check of the data, removing outliers that we view as being deleterious to the
180 interpretation. Shown in Figure 2 is an example of the impact that data cleaning has on the data quality. We
181 see that our process has removed spikiness in the data and smoothed out some of the small-scale perturbations,
182 resulting in a more manageable dynamic range.

183



184

185 Figure 2: An example of the measurement before (lower image) and after (upper image) data cleaning

186

187 *C. Machine Leaning Model Selection - Long Short-Term Memory (LSTM) vs. Autoregressive Integrated*
188 *Moving Average (ARIMA)*

189 In the choice of ML model to apply, we had to decide between a statistical approach based on ARIMA, or
190 alternatively, the use of a non-linear algorithm such as neural networks (NN). Many authors have investigated
191 comparing such methods and in various fields. What we noted from our review of the work was there was a
192 general agreement that approaches like ARIMA would not only require less inputs, but would be less of a
193 "black-box", which NNs are known to be [27, 28, 29].

194

195 ARIMA has been applied by previous authors to forecast oil production data [30]. The "autoregressive"
196 piece of ARIMA deals with finding a correlation between a specific value and a prior/lagged value, essentially
197 seeing if a variable has any correlation to its past values. The "integrated" piece deals with making data

stationary, essentially ensuring that properties of the data (such as mean and variance), are constant over time. The "moving average" piece of the model finds the dependency between a specific value, and the error from a moving average model applied to previous values. ARIMA models are therefore useful in forecasting time series data and are especially useful when trying to predict time series data that is non-stationary. While Ning et al observed that ARIMA was robust in predicting rates of oil production across wells, our review of ARIMA models being produced with high frequency data, where accuracy on an hourly basis was important, found that the error rate compounds significantly when the forecasting horizon is extended beyond a day [31].

LSTM is a type of recurrent NN which uses useful patterns from sequential data to provide accurate forecasts [20]. It learns from previous outputs to provide better results the following time. A typical LSTM has 3 layers (i) an input gate which assigns weights based on the significance of different variables, (ii) a forget gate to retain only useful information, and (iii) an output gate which manages the information flow. LSTM holds a memory cell (known as a cell state) which retains captured information over longer time periods and preserves useful constituents using its input and forget gates, hence avoiding the vanishing gradient issue associated with traditional NNs. LSTMs are particularly useful for non-linear problems where there does not appear to be strict mathematical relationships between variables. In our review of LSTM vs ARIMA models, we have found a common consensus in various fields that LSTM performed better, with reduced error rates but with significantly increased processing time [29, 28].

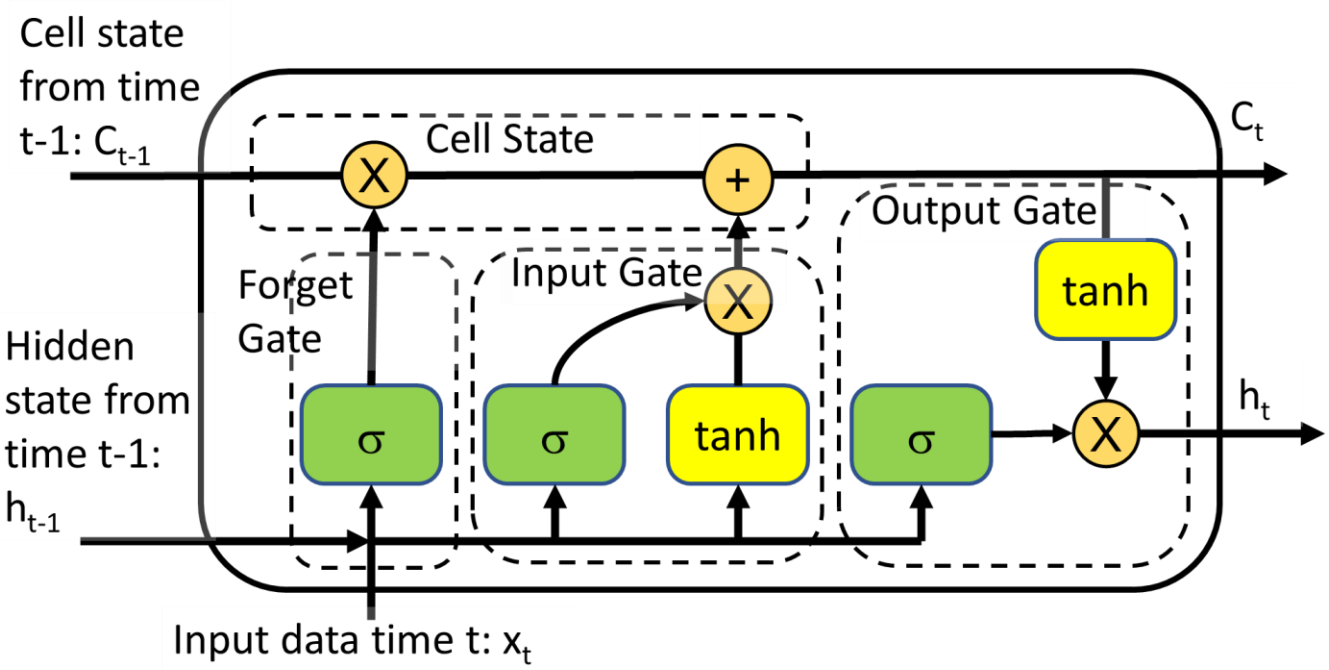


Figure 3: Schematic of LSTM network

We settled on the choice of LSTM because we realised that the data we were analysing was likely to contain non-seasonal, high frequency information, where accuracy was going to be important [32]. Additionally, the were numerous variables provided which we did not know the relative importance of in prediction without a working model.

D. Model Performance and Hyperparameter Optimization

We access the model performance using a host of measures, from training and validation loss to mean absolute error (MAE - Equation 4), mean squared error (MSE - Equation 5), root mean squared error (RMSE - Equation 6) and the coefficient of determination (R^2 - Equation 7). The training and validation loss functions serve to determine how well the model is performing and to prevent overfitting. During training, the model learns by iteratively adjusting its parameters to minimize a defined loss function. The training loss measures the discrepancy between the model's predictions and the actual target values on the training data. The goal is the minimization of this loss, as it indicates how well the model is fitting the training data. The validation loss is the other component of this and is computed by evaluating the model's performance on a separate validation dataset, not used for training. It serves as an estimate of how well the model generalizes to unseen data.

MAE: $$\frac{1}{n}\sum_{i=1}^{n}|y_i - x_i|$$ Equation 4

MSE: $$\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2$$ Equation 5

RMSE: $$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2}$$ Equation 6

R^2: $$1 - \frac{\sum(y_i - x_i)^2}{\sum(x_i - \bar{x})^2}$$ Equation 7

Equation 4 to Equation 7 show the other measurement metrics, with y_i as the predicted value, x_i as the true value, $\bar{x}$ as the mean of x_i and n as the total number of data points. MAE measures the average absolute difference between the predicted and actual values and provides an absolute measure of the model's performance. MSE measures the average squared difference between the predicted and actual values and amplifies larger errors due to the squaring operation. RMSE is the square root of MSE and provides a measure of the average magnitude of errors. It is useful for interpreting errors in the same units as the target variable. R^2 measures the proportion of the variance in the target variable that is explained by the model. It ranges from 0 to 1, where 1 indicates a perfect fit and 0 indicates a poor fit.

While MAE, MSE, RMSE, and R^2 are evaluation metrics used to assess the model's performance after training, the (training and validation) loss functions are specific to the training process. These loss functions guide the model's learning by providing gradients for updating the model's parameters. However, the suitability of the model is ultimately guided by scores where the MAE, MSE and RMSE and low, while the R^2 is high.

The NN model is tuned by varying a series of "hyperparameters". A hyperparameter is a characteristic of a model that is external to the model and whose value cannot be estimated from data. The hyperparameters that were optimized included the (i) learning rate, (ii) the epochs, (iii) the number of neurons, (iv) the

magnitude of dropout, (v) batch size, (vi) the choice of optimiser and finally (vii) the choice of activation function. To account for idiosyncrasies in the data (noise, patterns, outliers, etc.), k-fold cross-validation was run to validate the stability of the model.

Results and Discussion

Data Preparation

We observed significant collinearity of > 0.8 across most of the data set. Most were paired couplet measurements of "Temperature" and "Pressure" at various gauge depths. Keeping both parameters adds no additional information to the predictive model and in fact may be detrimental, with potential overfitting. Therefore, a single element from each of the variable pairs is eliminated to allow for a more stable model. Exceptions were made if variables were found to be from different sources e.g. tubing and reservoir pressure at the observation well for instance. This is also in line with conventional reservoir engineering concepts where temperature and pressure are correlated.

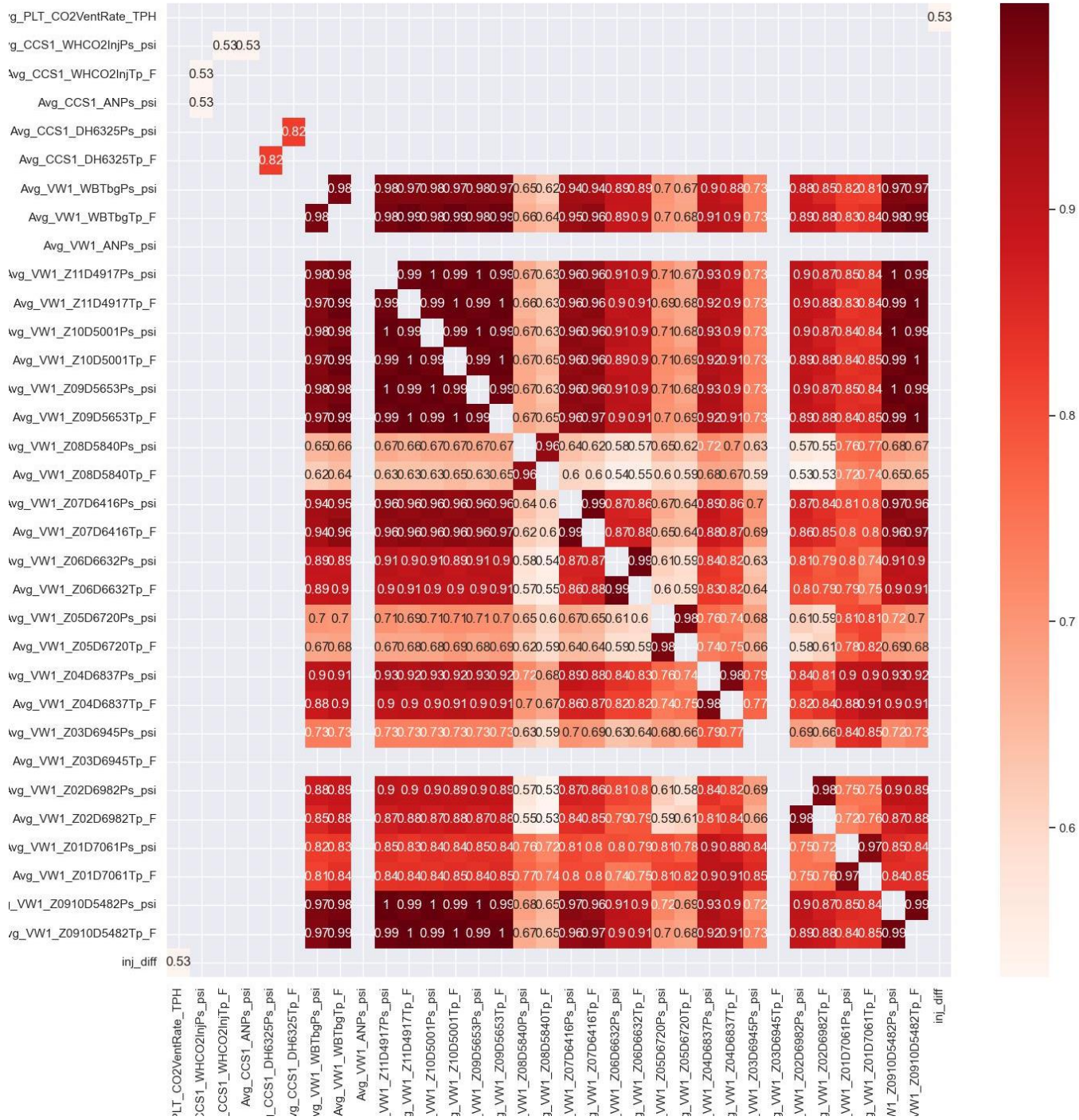


Figure 4: Correlation Matrix – Pairwise Correlation

Excluding the target injection delta variable, six variables were retained for the machine learning application and are summarised in Table 2 for the entire dataset. All data is numerical data types. The total data size is 27,665 values per column, for a total of 248,784 data points which was approximately ~26% of the original data base.

Table 2: Statistics of the cleaned dataset

Measurement	Non-Zero Value	Mean Value	Standard Deviation	Minimum Value	25th Percentile	Median Value	75th Percentile	Maximum Value
Avg_PLT_CO2VentRate_TPH	27,665	1.015	3.94	0	0	0.06	0.15	60
Avg_CCS1_WHCO2InjPs_psi	27,665	1253	191	701	1233	1339	1361	1882
Avg_CCS1_WHCO2InjTp_F		91	12	59	93	96	97	120
Avg_CCS1_ANPs_psi	27,665	554	102	149	523	565	605	977
Avg_CCS1_DH6325Ps_psi	27,665	3254	106	3006	3234	3286	3326	3516
Avg_CCS1_DH6325Tp_F	27,665	128	4.58	117	127	130	131	136
Avg_VW1_WBTbgPs_psi	27,665	2248	452	60	2233	2343	2421	4817
Avg_VW1_Z05D6720Ps_psi	27,665	3043	233	282	3061	3072	3075	3366
inj_diff	27,665	0.03	3.74	-38	-0.08	0	0.07	38

LSTM Model Architecture

We have employed a stacked LSTM architecture, with 2 layers, for this evaluation. In a stacked LSTM model, the output sequence of one LSTM layer serves as the input sequence for the next LSTM layer in the stack. This allows for a hierarchical representation of the input data, with each LSTM layer capturing different levels of abstraction or temporal dependencies. The second layer of the stacked model feeds the results to the output layer. We also use dropout to reduce the risk associated with overfitting. Finally, there is a dense layer which predicts the output, a single timestep at a time.

We tested the performance of the model by varying the hyperparameters given in Table 3. We utilised K-fold cross validation for hyper parameter tuning, ensuring that we kept the time series harmony. By this, what we mean is we split the time series sequence into samples but retained the sequence of information. With each split, a model is trained using (k-1) folds of the training data. The model is then validated against the remaining fold. A final model is scored on the held-out fold, with scores averaged across the splits. We used this to refine our hyperparameters, and finally took an average of values you see in Table 3.

Each hyperparameter serves a specific purpose and it is fundamentally an iterative process to tune these parameters such that the model outputs are optimised. We broadly define each hyperparameter here, sharing more details in Table 3. The same table also shows our final selected values based on an optimised MSE value.

The learning rate refers to the model's degree of responsiveness to errors. Epochs refer to the number of iterations across the entire dataset, while neurons are the fundamental nodes/building blocks used to process inputs. Dropout refers to the proportion of randomly selected neurons which get deactivated (to prevent overfitting) and batch size addresses the quantity of inputs processed before updating the model. An optimiser algorithm is needed to minimize the loss function, by finding the optimal set of parameter values that lead to improved network performance. Finally, the activation function is a mathematical function which introduces non-linearity to the layer of neurons.

Table 3: Variation in Hyperparameter Values, Final Selected Values and Parameter descriptor

Measure	Minimum	Maximum	Final	Parameter Descriptor
Learning Rate (lr)	0.001	0.1	0.01	The rate at which the parameters (weights and biases) of the network are updated during the training process. It essentially controls how quickly or slowly the network learns from the gradients of the loss function.
Epochs	10	100	50	This parameter determines how many times the learning algorithm will iterate over the entire training dataset. Each iteration has the following steps (a) forward propagation, (b) loss computation, (c) backpropagation, and (d) update. These steps are repeated in the training dataset until all samples have been processed. This completes one epoch.
Neurons	10	50	20	Each neuron takes multiple inputs, performs a computation on those inputs, and produces an output. The computation typically involves applying a weighted sum of the inputs, followed by the application of an activation function.
Dropout	0.1	0.25	0.25	This parameter introduces noise or randomness into the network by temporarily removing a portion of the neurons from the calculation in each

				training iteration. By doing so, dropout prevents complex co-adaptations between neurons, reducing the reliance of the network on specific neurons and promoting the learning of more robust and generalized representations.
Batch Size	30	100	50	This parameter refers to the number of training examples that are processed together in a single forward and backward pass during training.
Optimiser	-	-	Adam	ADAM (Adaptive Moment Estimation) is an optimization algorithm used to update the weights of neural networks during the training process. It automatically adjusts the learning rate for each parameter based on the history of gradients. This was chosen as the default optimizer as it has the ability to converge quickly as well
Activation	-	-	tanh	The hyperbolic tangent function maps the input to a value between -1 and 1. It has an S-shaped curve like the sigmoid function but is symmetric around the origin. The formula for tanh is: $f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$

Figure 5 shows the MSE output of the training (train) and validation (val) sets for one of our K-folds. MSE is on the y-axis and epoch is on the x-axis. We observe the convergence between train and val, telling us that the model accuracy is improving per epoch. The point of convergence is arrived at by gradient descent and tells us the point of minimum information loss. In this example, we achieved convergence at around epoch = 30, with losses increasing beyond that.

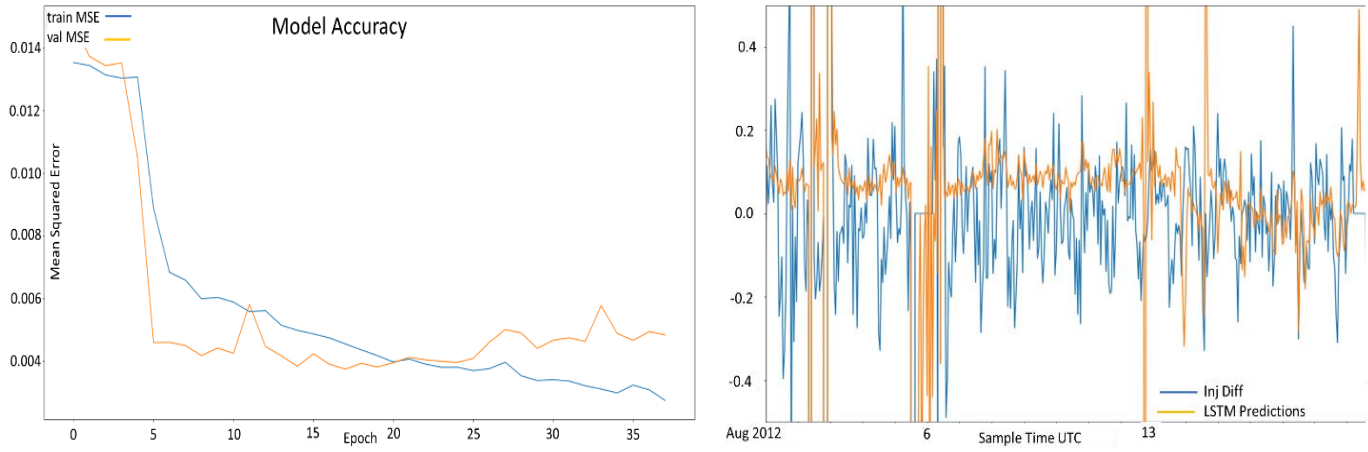


Figure 5: (a) MSE output of the training and validation set (b) Check file of a small subset of data; output from the K-fold set.

Prediction on Validation Data

We deployed our developed model on our hold out data. Given in Figure 6 is the comparison between the predicted injection rate deltas versus the actual injection rate deltas in our validation set. The results demonstrate the model's capability to successfully identify and replicate actual anomalies, which provides evidence that the model is predictive in nature. Moreover, Figure 6 illustrates that the model is able to accurately capture both large scale high frequency signal, as well as smaller scale fluctuations in the injection deltas, indicating the robustness of the LSTM model.

The fact that the model can capture both large and small-scale measures strengthens our view that the model is indeed reliable in handling the variety of conditions experienced in the field during CO_2 injection, and hints at its ability to effectively mimic the intricate patterns found within field injection rate data. The successful reproduction of significant anomalies as well as subtle variations highlights the model's potential to provide valuable insights and reliable predictions in this domain.

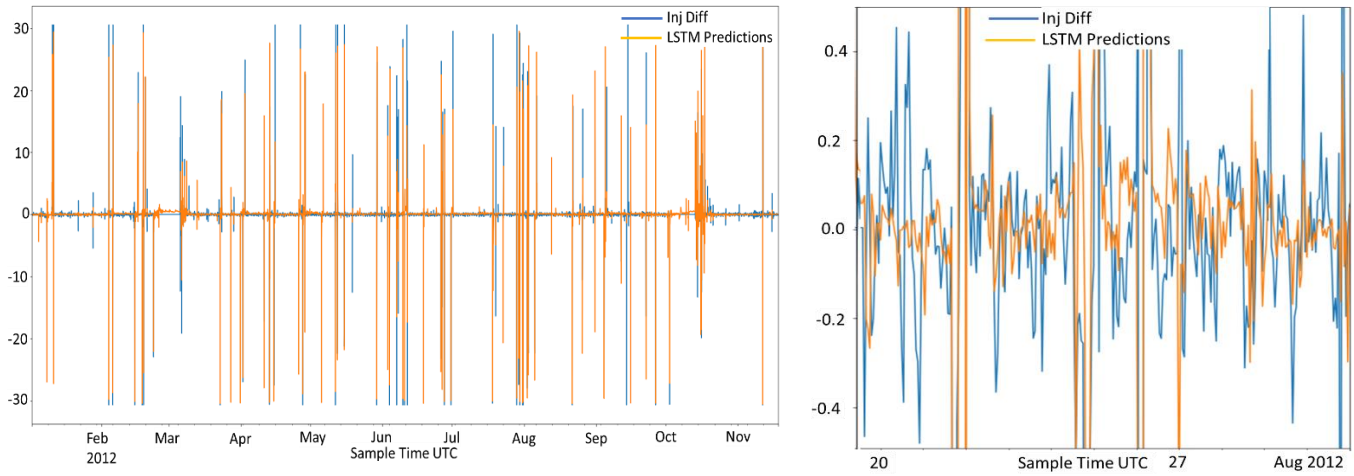


Figure 6: (a) Able to match anomalies in injection delta (b) Able to mimic the small variations in the injection delta.

Generalising the model

This LSTM model was built from (what is fundamentally) a single data set i.e. from a specific geological rock type, whose CO_2 flow behaviour is constrained by its own petrophysical and reservoir engineering properties. We think further generalisation of the model is necessary to truly confirm these findings. As we do not have data from other geological settings, we attempted generalisation through a series of sensitivity runs. We did this in 2 ways (a) selecting a single seed value but varying the input parameters fed into the model and (b) by randomising the seed value itself but keeping the input parameters constant.

(a) Single Seed, Variable Inputs

We used a single seed value (2250) but varied six of the selected as a test for model generalisation. Our first iteration (Run 1 in Table 4) is with a default hyperparameter setting, where the predicted injection delta is zero for all test timesteps; this yielded an RMSE value of 5.54.

Several adjustments were made to the inputs from that base case defined in run 1. These modifications included (a) scaling all columns within a range of -1 to 1 (runs 2 to 6), (b) replacing, as one of the input variables, the VW DH Z09 sensor with the VW DH Z05 sensor (runs 4 to 6), (c) incorporating well head pressure (WHP) sensor data (runs 2 to 4, run 6), (d) using temperature sensor data (runs 5 and 6), (e) increasing the Z-score band for the target column from 20 to 25 (runs 3 to 6), and (f) utilizing corrected values of zero for the VW DH sensor (runs 3 to 6).

By analysing the results of these runs and comparing them to the base and default settings, it is possible to determine the most optimal configuration that yields improved accuracy and reliability in predicting injection deltas. This series of sensitivity runs showed us that the model performs most optimally based on the configuration in run 6. It additionally tells us that the model is sensitive to scaling, and our Z-score tolerance bands.

Table 4: RMSE and R2 value of sensitivity runs to test randomness of model

Varied Parameter	Run 1 (Base Case)	Run 2	Run 3	Run 4	Run 5	Run 6 (Final Model)
Scaling	N	Y(0,1)	Y(0,1)	Y(-1,1)	Y(-1,1)	Y(-1,1)
VW DH Sensor	NA	Z09	Z01	Z05	Z05	Z05
Injection WHP Sensor	NA	Y	Y	Y	N	Y
Temp Sensor	NA	N	N	N	Y	Y
Z-Score Inj_Diff	NA	20	25	25	25	25
VW Zero Values	NA	Yes	No	No	No	No
Test RMSE	5.54	2.65	1.82	1.98	1.50	1.31
Val RMSE	NA	1.75	2.20	1.90	2.01	1.73
Val R2	NA	0.73	0.63	0.72	0.69	0.77

Randomness

NN themselves are not inherently stochastic, as they follow deterministic mathematical operations, and the inputs, weights, and biases are all fixed values during inference. However, the stochastic nature of the NN model becomes apparent when seed values are varied; different seed values result in different results.

Our next test for the generality of the model involved running various seed values (we utilised 10 different values) and determining if both high and low amplitude delta values could be predicted. Six selected results are displayed in Table 5 while Figure 7 to show the range of comparison with respect to the first unseeded run.

We observe that while all runs can predict the anomalies, the signature of the minor differences varies between seed values. We also observe that the RMSE and R2 values on the validation set all lie within a small range (±10%) of the unseeded value. Thus, to account for randomness and to retain the unbiased nature of the model, an unseeded model is selected for deployment.

Table 5: RMSE and R2 value of sensitivity runs to test randomness of model

Seed Value	Unseeded	2250	42	2023	1	111	88
Val RMSE	1.73	1.55	1.69	1.69	1.64	1.87	1.70
Val R2	0.77	0.82	0.78	0.78	0.79	0.73	0.78

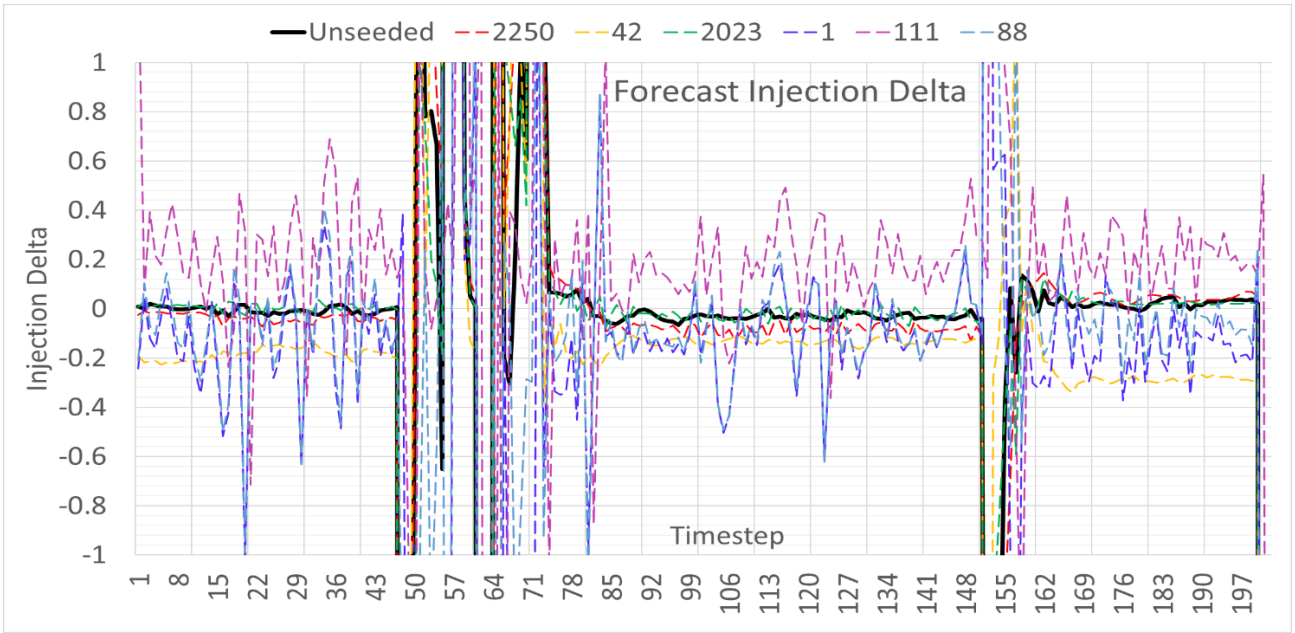


Figure 7: Injection Delta forecasts of sensitivity run to test randomness of model – all results show small perturbations around the unseeded value

Evaluation against the "Hold-out" Data

Figure 8 shows the model inputs and prediction on the "hold out" dataset. The result shows that there are four anomalies in the predicted injection deltas.

When correlated back to the input dataset, we observe that for the negative injection deltas, there are corresponding decreases in injection pressure implying a decreased CO_2 injection rate.

The prediction data also shows corresponding increase in injection pressure during positive injection deltas, implying an increased injection rate. Corresponding changes can also be seen on the other inputs such as temperature and CO_2 vent rate.

Discussion & Recommendations

We think there are several ways to further enhance the model's performance. First, incorporating the response of the operator when an anomaly is marked as safe reduces the likelihood of flagging similar anomalies in the future. This feedback loop mechanism can help refine the model's predictive accuracy and ensure more reliable results over time.

Secondly, exploring alternative methods for outlier removal, and implementing advanced techniques to identify and handle outliers effectively will improve the model's ability to identify genuine anomalies, while minimizing false positives.

Thirdly, the workflow of the model can be improved to reduce runtime. This optimization involves running multiple sets of model ensembles, and aggregating their outputs to obtain a more "generalised" model. Generalising the model can further enhance performance and provide more reliability when it comes to anomaly detection.

Conclusion

The IBDP project involved injection of ~1000 tonnes/day of CO_2 into a single well, for a three year, resulting in ~995, 215 cumulative tonnes of CO_2 being injected. In view of the growing importance of CCUS in achieving climate targets, we believe the wealth of open-source time-series data from this project can be applied to derisk CCUS monitoring.

In this work, we demonstrated how an LSTM based machine learning model has been developed and implemented to predict injection deltas of active CO_2 injection. We demonstrate that our predicted models do well based on RMSE and R2 scores against both the training and validation data sets. Our model was able to

predict both large and small anomalies, and we demonstrated that the model is sufficiently generalised for this single input data source.

The primary objective of this model is to detect anomalies and alert operators to closely inspect the wells for potential leaks. This predictive capability becomes especially valuable in scenarios where multiple wells are in operation, as it eliminates the need for additional operators, thereby reducing operational costs. We also recommended a series of improvements and model enhancements in this paper which would increase the machine learning model's predictive ability, potentially providing operators with timely alerts for potential leaks while optimizing operational efficiency and cost-effectiveness.

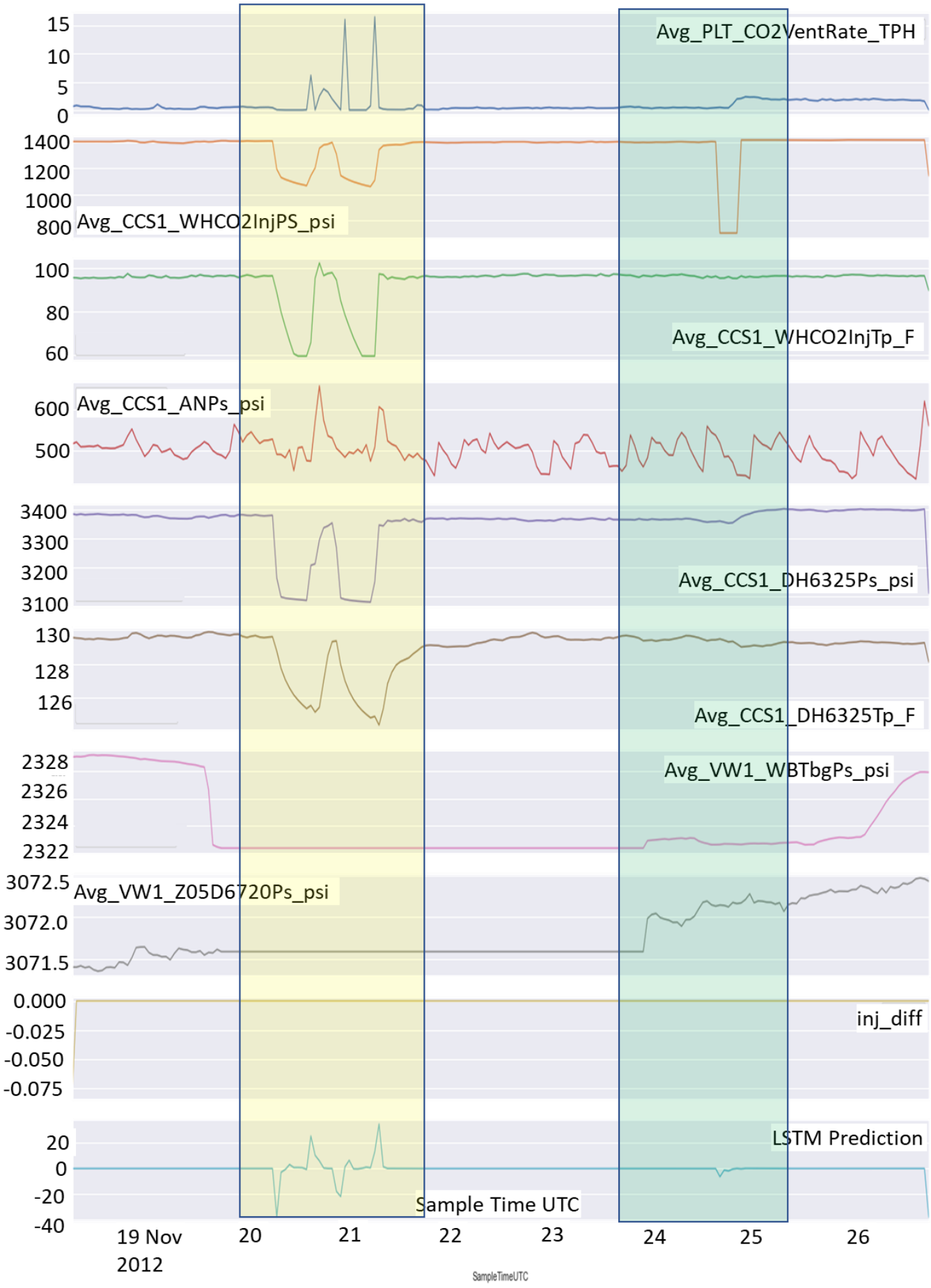


Figure 8: Input and prediction on "hold out" dataset

Nomenclature

GHG	greenhouse gas
CO_2	carbon dioxide
CH_4	methane
N_2O	nitrous oxide
EPA	Environmental Protection Agency
CCUS	Carbon Capture Utilisation and Storage
EOR	Enhanced Oil Recovery
Δ	injection rates deltas
IR	injection rate
ML	machine learning
ARIMA	Autoregressive Integrated Moving Average
ARFIMA	Autoregressive Fractionally Integrated Moving Average
ARCH/GARCH	Autoregressive Conditional Heteroskedasticity/ Generalized Autoregressive Conditional Heteroskedasticity
DCA	decline curve analysis
q_i	initial rate (bbls/day)
D_i	initial decline rate (units)
b	degree of curvature of the line
AI	artificial intelligence
LSTM	Long Short-Term Memory
IBDP	Illinois Basin Decatur Project
TD	total depth
MAD	median absolute deviation

NN	neural networks
MAE	mean absolute error
MSE	mean squared error
RMSE	root mean squared error
R^2	coefficient of determination
lr	Learning Rate
ADAM	Adaptive Moment Estimation

Acknowledgements

The authors wish to acknowledge Ryan Lazaroo for peer review of this work.

Reference

[1] R. Lindsey, "Climate Change: Atmospheric carbon dioxide, NOAA Climate.gov," [Online]. Available: https://www.climate.gov/news-features/understanding-climate/climate-change-atmospheric-carbon-dioxide. [Accessed 25 April 2023].

[2] United States Environmental Protection Agency, "Sources of Greenhouse Gas Emissions," 20 Jul 2022. [Online]. Available: from https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions.

[3] R. A. W. R. G. S. E. &. W. S. G. Bauer, "Illinois basin–Decatur project.," in *Geophysics and geosequestration*, Cambridge University Press, 2019, pp. 339-369.

[4] J. G. D. Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International Journal of Forecasting*, vol. 22, no. 3, pp. 443-473, 2006.

[5] E. S. Gardner, "Exponential smoothing: The state of the art," *Journal of Forecasting*, vol. 4, no. 1, pp. 1-28, 1985.

[6] R. D. Snyder, "Recursive Estimation of Dynamic Linear Models," *Journal of the Royal Statistical Society. Series B (Methodological),* vol. 47, no. 2, p. 272–276, 1985.

[7] R. H. Scumway and D. S. Stoffer, Time Series Regression and ARIMA Models, New York: Springer, 2000.

[8] G. Huyot, K. Chiu, J. Higginson and N. Gait, "Analysis of Revisions in the Seasonal Adjustment of Data Using X-11-Arima Model-Based filters," *International Journal of Forecasting,* vol. 2, no. 2, pp. 217-229, 1986.

[9] S. H. Babbs and K. B. Nowman, "Kalman filtering of generalized Vasicek term structure models," *Journal of financial and quantitative analysis,* vol. 34, no. 1, pp. 115-130, 1999.

[10] M. P. F. P. H. &. S. N. R. Clements, "Forecasting economic and financial time-series with non-linear models," *International journal of forecasting,* vol. 20, no. 2, pp. 169-183, 2004.

[11] B. K. Ray, "Long-range forecasting of IBM product revenues using a seasonal fractionally differenced ARMA model," *International Journal of Forecasting,* vol. 9, no. 2, pp. 255-269, 1993.

[12] S. J. Taylor, "Forecasting the volatility of currency exchange rates," *International Journal of Forecasting,* vol. 3, no. 1, pp. 159-170, 1987.

[13] T. R. Willemain, C. N. Smart, J. H. Shockor and P. A. DeSautels, "Forecasting intermittent demand in manufacturing: a comparative evaluation of Croston's method," *International Journal of Forecasting,* vol. 10, no. 4, pp. 529-538, 1994.

[14] J. J. Arps, "Analysis of Decline Curves," *Transactions of the AIME,* Vols. SPE-945228-G, 1945.

[15] A. L. L. a. T. P. Clark, "Production Forecasting with Logistic Growth Models," in *SPE Annual Technical Conference and Exhibition*, Denver, Colorado, USA, 2011.

[16] A. Duong, "An Unconventional Rate Decline Approach for Tight and Fracture-Dominated Gas Wells," in *Canadian Unconventional Resources and International Petroleum Conference. ,* Calgary, Alberta, Canada, 2010.

[17] D. R. J. A. P. A. D. a. T. A. B. Ilk, “Exponential vs. Hyperbolic Decline in Tight Gas Sands — Understanding the Origin and Implications for Reserve Estimates Using Arps' Decline Curves.,” in *Paper presented at the SPE Annual Technical Conference and Exhibition*, Denver, Colorado, USA, 2008.

[18] P. Valko, “Assigning value to stimulation in the Barnett Shale: a simultaneous analysis of 7000 plus production hystories and well completion records,” in *SPE Hydraulic Fracturing Technology Conference* , Woodlands, Texas. , 2009.

[19] M. Kumar, K. Swaminathan , A. Rusli and A. Thomas-Hy, “Applying Data Analytics & Machine Learning Methods for Recovery Factor Prediction and Uncertainty Modelling,” *SPE Asia Pacific Oil & Gas Conference and Exhibition,* Vols. SPE-210769-MS , pp. 1-12, 2022.

[20] N. Ibrahim, A. Alharbi, T. Alzahrani, A. Abdulkarim, I. Alessa, A. Hameed, A. Albabtain, D. Alqahtani, M. Alsawwaf and A. Almuqhim, “Well Performance Classification and Prediction: Deep Learning and Machine Learning Long Term Regression Experiments on Oil, Gas, and Water Production,” *Sensors,* vol. 22, no. 5326, pp. 1-22, 2022.

[21] P. Panja, W. Jia and B. Mcpherson, “Prediction of well performance in SACROC field using stacked Long Short-Term Memory (LSTM) network,” *Expert Systems With Applications* , vol. 205, no. 117670, pp. 1-25, 2022.

[22] Y. Dong, Y. Zhang and X. Cheng, “Reservoir Production Prediction Model Based on a Stacked LSTM Network and Transfer Learning,” *ACS Omega,* vol. 6, pp. 34700-34711, 2021.

[23] P. Yao, Z. Yu, Y. Zhang and T. Xu, “Application of machine learning in carbon capture and storage: An in-depth insight from the perspective of geoscience,” *Fuel*, vol. 333, no. 1, p. 126296, 2023.

[24] B. Chen, D. R. Harp, Y. Lin, E. H. Keating and R. J. Pawar, “Geologic CO2 sequestration monitoring design: A machine learning and uncertainty quantification based approach,” *Applied Energy*, vol. 225, no. 1, pp. 332-345, 2018.

[25] J. You, W. Ampomah, Q. Sun, E. J. Kutsienyo, R. S. Balch, Z. Dai, M. Cather and X. Zhang, "Machine learning based co-optimization of carbon dioxide sequestration and oil recovery in CO2-EOR project," *Journal of Cleaner Production,* vol. 260, p. 120866, 2020.

[26] U. P. Iskander and M. Kurihara, "Long Short-term Memory (LSTM) Networks for Forecasting Reservoir Performances in Carbon Capture, Utilisation, and Storage (CCUS) Operations," *Scientific Contributions Oil & Gas,* vol. 45, no. 1, pp. 35 - 50, 2022.

[27] M. Alsharif, M. Younes and J. Kim, "Time Series ARIMA Model for Prediction of Daily and Monthly Average Global Solar Radiation: The Case Study of Seoul, South Korea.," *Symmetry,* vol. 11, no. 2, p. 240, 2019.

[28] S. Siami-Namini, N. Tavakoli and A. S. Namin, "A comparison of ARIMA and LSTM in forecasting time series," *17th IEEE international conference on machine learning and applications (ICMLA),* pp. 1394-1401, 2018.

[29] S. A. Ludwig, "Comparison of time series approaches applied to greenhouse gas analysis: ANFIS, RNN, and LSTM.," *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE),* pp. 1-6, 2019.

[30] Y. Ning, H. Kazemi and P. Tahmasebi, "A comparative machine learning study for time series oil production forecasting: ARIMA, LSTM, and Prophet," *Computers & Geosciences,* vol. 164, no. 105126, 2022.

[31] A. Parasyris, G. Alexandrakis and G. Kozyrakis, "Predicting Meteorological Variables on Local Level with SARIMA, LSTM and Hybrid Techniques," *Atmosphere,* vol. 13, no. 878, 2022.

[32] G. Reikard, "Predicting solar radiation at high resolutions: A comparison of time series forecasts," *Solar Energy,* vol. 83, no. 3, p. 342–349, 2009.

436

437

438

Declaration of interests

☒The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: