UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

AUTOMATIC LEAK DETECTION IN CARBON SEQUESTRATION PROJECTS

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

Saurabh Sinha

Norman, Oklahoma

2020

AUTOMATIC LEAK DETECTION IN CARBON SEQUESTRATION PROJECTS

A THESIS APPROVED FOR THE

GALLOGLY COLLEGE OF ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Talayeh Razzaghi

Dr. Shima Mohebbi

Dr. Kurt J. Marfurt

Contents

List of Tables

List of Figures

Acknowledgments

As a graduate student at the University of Oklahoma from past six years, I had the opportunity to work in petroleum engineering and geophysics. As I was eager to learn about the emerging technology in the field of machine learning and data science, I decided to pursue a concurrent master in data science as well. I would like to thank my PhD guide Dr. Kurt Marfurt for his support during this period and encouraging me to pursue the data science degree and providing funding for my study in this period. As my work is multidisciplinary in nature, it requires expertise from both the data science and the geophysics. I would like to express my gratitude towards my committee chair Dr. Talayeh Razzaghi and Dr. Shima Mohebbi for providing me valuable inputs for this project from a data science perspective. Recently, I had an opportunity to join Los Alamos National Laboratory (LANL) in January, 2020 as an intern. The data used in this study is provided by my mentor at LANL, Dr. Lin Youzuo. I would like to wholeheartedly thank Dr. Youzuo for giving me an opportunity to work in state of the art technology available today in the field of applied machine learning in geophysics and guiding me in every aspect of the project. I would also like to thank United States Department of Energy (DOE) for providing funding for this project. Last but not the least, I would like to thank Dr. Rafael Lima and Yuliana Zapata for their support during this whole period as the best friends I could have ever asked for.

Abstract

The international commitments for carbon capture will require a rapid increase in carbon capture and storage (CCS) projects. The key to any successful carbon sequestration project lies in the long term storage and prevention of leakage of stored carbon dioxide (CO_2). In addition to being a greenhouse gas, CO_2 leaks reaching the surface can accumulate in low-lying areas resulting in a serious health risk. Among several alternatives, some of the more promising CSS storage formations are the hundreds of thousands of depleted oil and gas reservoirs, where by definition the reservoirs had good geological seals prior to hydrocarbon extraction. With more CSS wells coming online, it is imperative to implement permanent, automated monitoring tools. In this study we applied machine learning models to automate the leakage detection process in carbon storage reservoirs using rates of supercritical (CO_2) injection and pressure data measured by simple pulse tests. To validate the promise of this machine learning based workflow, we implemented data from pulse tests carried out in the Cranfield reservoir, Mississippi, USA. The data consist of a series of pulse tests conducted with baseline parameters and with an artificially introduced leak. Here, we pose the leakage detection task as an anomaly detection problem where deviation from the predicted behavior indicates leaks in the reservoir. The results obtained show that different machine learning architectures such as multi-layer feed forward network, Long Short-Term Memory , convolutional neural network are able to identify

leakages and can flag an early warning. These warnings can then be used by human interpreters to take remedial measures.

Introduction

Carbon capture and storage (CCS) refers to the process of permanently capturing the CO_2 emitted primarily by electricity generation, cement processing, and other fixed industrial assets into geological formations (Selma et al., 2014). Depleted hydrocarbon reservoirs serve as an excellent choice for CCS (Jan et al., 2017; Bachu, 2000, 2003) due to their higher storage capacity with available infrastructure in place. Older hydrocarbon production/injection wells can be re purposed as injection wells for the CO_2 injection.

Although complex, the physics of fluid flow in geological reservoirs is fairly well understood by petroleum engineering community. CO_2 injection for enhanced oil recovery (EOR) projects has been used for decades in the oil industry. At atmospheric conditions, CO_2 exits as a gas with with a density of 1.872 kg/m3. During the high pressure injection, density of CO_2 increases causing phase changes (Figure 1). Note in (Figure 1) when the pressure and temperature exceed beyond critical point (Hoteit et al., 2019), CO_2 exists in supercritical phase which is a desirable condition.CO_2 in supercritical state exhibits higher densities and hence more volume of CO_2 can be injected. All CCS projects exploit this property of CO_2 for injection.

The injected CO_2 can take a significant time to convert into stable form by either dissolution, precipitation etc. (Gaus, 2010; Oelkers et al., 2008; Moore et al., 2005). Hence, key to any successful CCS project is the long term storage of the CO_2

without leaks. Leakage, will not only negate the purpose of a CCS project, the injected CO_2 can migrate to the surface and pose a different set of environmental problems. Leaked CO_2 can mix with the water table and increase its acidity of the water table, it can also affect severely plant growth at surface soil levels, soil microbiology (Smith et al., 2013; Fernández-Montiel et al., 2015).

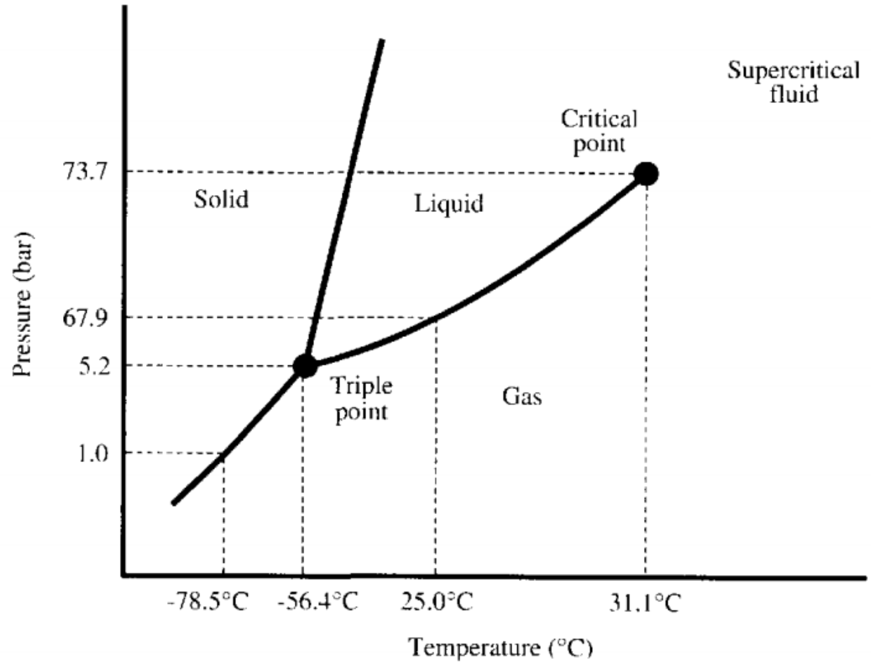

Figure 1. *CO_2 phase behaviour. Notice at higher pressure and temperature CO2 behaves as a supercritical fluid which is often the case with CCS projects (Rochelle et al., 2004).*

The process of CO_2 injection is not just operationally challenging but also time dependent. In a CCS project, CO_2 is injected into pressure-depleted hydrocarbon reservoirs, increasing the reservoir pressure and hence changing the stresses in the reservoir and potentially reactivating faults in the area (Castelletto et al., 2013; Rutqvist, 2012; Ivanova et al., 2012). If the faults slip, the stored CO_2 in the reservoir may be able to migrate to the surface.

The detection of CO_2 leakage in CCS project can be assessed via several technologies. Gal et al. (2019) and May and Waldmann (2014), measured CO_2 concentrations in surface soils. Shao et al. (2019a) found carbon isotopes in the soil served as a proxy for the CO_2 leak, whereas Shao et al. (2019b) showed that surface measurements of tracers injected with the CO_2 can indicate leaks. Although, most of these methods are economically viable and are direct indicator of CO_2 leak, they do not provide an early warning signal to prevent the leak during the operational stage. By the time the leak is detected, a significant amount of CO_2 may already have reached to the surface.

Two reliable and well-established field methods for continuous monitoring of subsurface properties are via seismic monitoring and the hydrocarbon well pressure monitoring. Stork et al. (2018), Macquet et al. (2017), Roach et al. (2015) and, Bergmann et al. (2010) showed the usage of seismic data in leak detection in the hydrocarbon reservoir. Three-dimensional (3D) seismic data provide a 3D image of the changes in velocities and density caused by a $\mathrm{CO_2}$ plume. A time-lapse analysis of 3D seismic can provide valuable insights on plume development and migration. Figure- 2 shows one such study from the Sleipner field in the North Sea (Chadwick et al., 2009). Notice the "bright spots" in seismic amplitude after eight years of continuous CO_2 injection (Johnston, 2013).

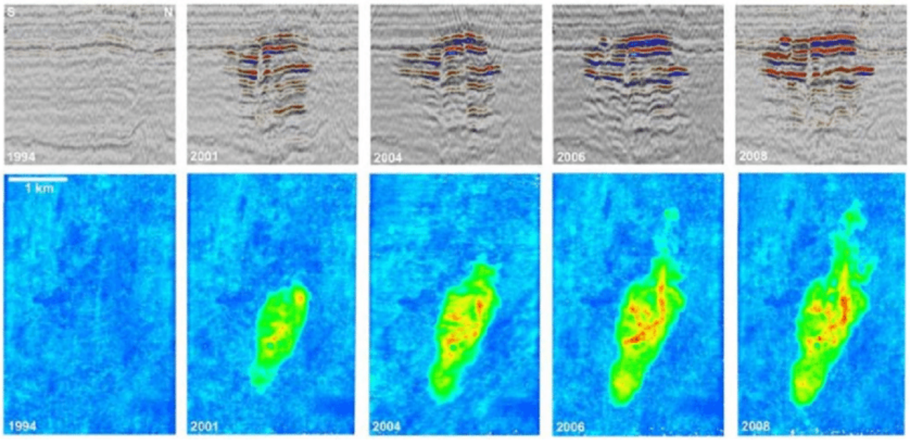

Figure 2. *Time lapse seismic acquired from Chadwick et al. (2009) showing the CO_2 plume development on seismic amplitude in Sleipner field in the North Sea between 1994 to 2008.The corresponding density changes are recorded as "bright spots" in the amplitude data.*

The expert interpretation of the seismic data is expensive and time-consuming even for passive monitoring (Verdon et al., 2010a,b; Dondurur, 2018). Additionally, seismic data are often band limited providing relatively low vertical resolution images, and hence an only approximate estimate (e.g. within 50-100 m) on the spatial location of any leak. The quality of the seismic interpretation is dependent on data acquisition, data processing, and the initial guess of the geological model itself. The data acquired from a single geophone is a measurement obtained in the form of a time series data. The migrated seismic amplitude data however, is obtained by complex processing with best guesses of geology and velocity models (Olivier et al., 2018; de la Iglesia et al., 2018). The fluid substitution models used for the time-lapse analysis, pose its own set of limitations such as assumptions about the fluid properties (Moradi and Lawton, 2013).

An alternative to the time-lapse seismic data monitoring is pressure data monitoring. The formation pressure is measured with a pressure gauge which is both inexpensive and easy to install. The pressure gauge is installed at the desired target formation of a well. The gauge continuously provides high frequency, high resolution data in real time via a telemetry system (Hawthorn et al., 2017; Reeves et al., 2011). The injection of CO_2 causes a pressure perturbation and can be induced intentionally in a process called a ”well test”. One such well test is a pulse test.

A typical pulse test consists of alternative cycles of fluid injection and well shut-in causing multiple pressure perturbations or sampling the reservoir at multiple frequencies (Hosseini, 2014; Brigham et al., 1970; Fokker and Verga, 2011; Fokker et al., 2018). The pressure data are obtained at the bottom of the well and hence exhibit a high signal to noise ratio.

The pulse test typically requires at least two wells. The first is an injection well at which this perturbation is induced by the harmonic injection of a fluid into the reservoir. The second is an observation well which records the response of the reservoir due to these pulses. Analyzing the pressure signal at the observation well provides various insights about the reservoir and the fluid properties such as reservoir connectivity, fault proximity, permeability etc. Sun et al. (2016) demonstrated the efficacy of the pressure pulse test in distinguishing the pressure response of a leak versus the non-leak case in a field test in Cranfield, Mississippi, USA. Section 4 of this thesis discusses the details of the pulse test. The fact that a pulse test can distinguish between the leak and no-leak scenarios lay the foundation of this work.

Analysis of pulse test by human interpreters is not an extremely complicated task. A skilled human interpreter can easily distinguish the difference between leak and non-leak responses. However, a large CCS project across multiple depleted oil fields may incorporate thousands of injection wells. Each well will be instrumented, with pulse tests being conducted every year. Continuous analysis of so many live pressure feeds can be a challenging task for human interpreters. In contrast, computers are extremely efficient at repetitive tasks and intuitively, are best suited for this purpose.In contrast, computers are extremely efficient at repetitive tasks and intuitively, are best suited for this purpose. Combined with the fact that a pulse test can distinguish between a leak and non-leak scenarios and that process can be automated is the core of my work.

The use of machine learning in CO_2 leakage detection is not a novel idea. Zhou et al. (2019) demonstrated the use of convolutional neural networks(CNN) with synthetically modeled seismic data to monitor the CO_2 volume. Zhong et al. (2019b) showed the use of reservoir simulator images for a convolutional-Long Short-Term Memory (Conv-LSTM architecture). Sun et al. (2014a) proposed segregating the leak and the baseline pressure response using CNN on spectral domain images. All of the above methods are either based on seismic data or two-dimensional images which either suffers from resolution issues or is computationally expensive. In many cases, the data required for simulation or the tool itself is not available at field limiting the practicality of these methods.

The aim of this work is to create an early warning detection system for leakages in CCS projects using pressure data along with rate and cumulative injection volume of supercritical CO_2. We used currently available state-of-the-art deep learning models and tune them to suit the data. The data consists of time series data of injection rates and measured pressures from a set of pulse tests conducted at the Cranfield reservoir, Mississippi, USA. As we do not utilize any two-dimensional images or three-dimensional voxels in this study, the methodology is computationally efficient while providing state-of-the-art results.

We implemented deep learning architectures such as multi-layer feed forward

(MFNN), Long Short-Term Memory (LSTM), convolutional neural networks (CNN) and a combination of CNN and LSTM, abbreviated as CONV-LSTM. The results show that if trained on a particular set of data, a machine can identify anomalous behavior from the learned data and flag it as an anomaly. Sophisticated architectures such as CNN and LSTM can learn and retain features present in typical pulse test pressure response. We also show the adaptability of the models for different pulse test as well as its limitations. The results can be achieved in a fraction of time using simple inputs but, sophisticated architectures. The final product of this analysis is an anomaly detector which can differentiate between expected versus the anomalous pressure behavior.

This thesis is divided into five sections. Section one provides information about the data used in this manuscript, the problem formulation and data pre-processing for our workflow. Section two provides a basic introduction of the neural network architectures used and the tuned version used for analyses. Section three contains results from the neural network models and the discussion of these results. Section four describes the novelty of the work and the comparisons to the current state of the art. Section five presents the thesis conclusions.

Background

Cranfield site data acquisition

The data utilized in this study was obtained in the Cranfield reservoir, Mississippi, USA. The Cranfield reservoir is primarily a sandstone reservoir which was used for CO_2 enhanced oil recovery project after the primary and secondary recovery (Sun et al., 2016). The CO_2 injection for sequestration purpose started in July 2008. The pulse test used in this study was carried out in January 2015, the total injected volume into the reservoir at the time of the test was more than one million metric tons. Due to the high injection volumes continuously for seven years prior to the test, it is assumed that in-situ brine effects are negligible for any practical purposes and supercritical CO_2 is the only in-situ fluid present in the reservoir.

Figure 3 shows the setup used in the pulse test as described by Sun et al. (2016). Three wells are utilized in the study F1, F2, and F3. The distance between the wells is also shown in the Figure 3. The CO_2 is injected in well F1 which is the injection well, well F2 is the monitoring well, and F3 is a well that can be used to add an artificial leak by venting off CO_2 via a surface valve to simulate a real life leakage scenario.

Hereinafter, we refer as baseline, the cases on which the pulse tests are conducted without the introduction of a leak, and an alternating sequence of injection and shut-in is carried out. Two baseline tests, one of 90-minutes and one of 150-minutes, were conducted on 19th January and January 20th of 2015, respectively. Leak tests are similar to baseline tests, but with an artificial leak introduced at well F3 by opening a surface valve. Two leak tests were conducted ten days after the baseline tests, one on 30th of January 2015 and another on 31st of January 2015 with a duration of 90-minutes and 150-minutes respectively. Well F2 remains shut-in as an observation well during all tests and pressure data acquired at well F2 is used in our analysis. Table 1 summarizes the pulse test schedule for all four tests. Figure 4 shows the measured pressure recorded at well F2 during the test, along with the injection rates at well F1. We only provided the necessary details about the data

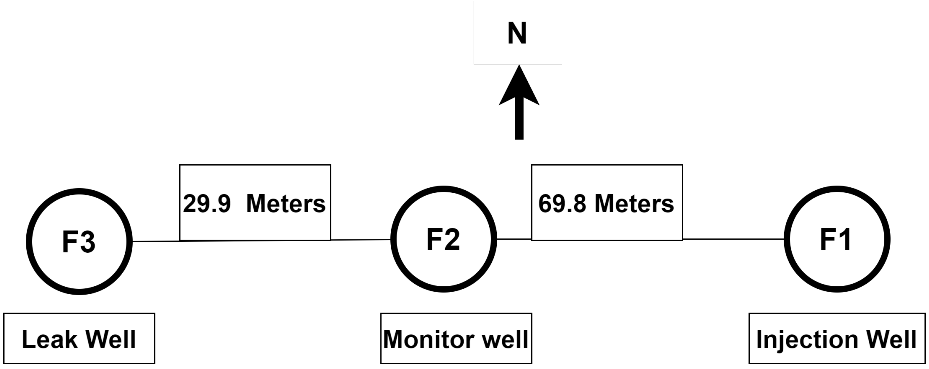

Figure 3. *Schematic illustration of the wells used in this study and their configuration. F1 is the injection well, F2 is the monitoring well and F3 is the well where leak is introduced. Pressure utilized in this study is obtained from the well, F2.*

Test	Date	Rate	Pulse
Baseline - 90 min	19th Jan 2015	3,620	45
Baseline - 150 min	20th Jan 2015	3,620	75
Leak - 90 min	30th Jan 2015	3,620	45
Leak - 150 min	31st Jan 2015	3,620	75

Table 1. *Injection schedule for the baseline and leak test. Pulse duration is identical in respective baseline and leak tests. All rates are measured in Bbl/D and the pulse half cycle times are in minutes.*

for the objectives of this study. A complete description of the data acquisition is provided by Sun et al. (2016).

The pressure gauge sampling rate used in the baseline tests is one sample every two seconds i.e., a sampling rate of 0.5 samples/second. The sampling rates used in the leak tests are one sample every five seconds i.e., a sampling rate of 0.2 samples/ second. Before any analysis, we matched the sampling rates of both baseline and leak data to 1 sample/second. As very limited data is available to train the models, we used the Fourier method to re-sample the signal to 1 sample/second instead of decimating the signal (Heideman et al., 1984). The data pre-processing steps are summarized in Figure- 5.

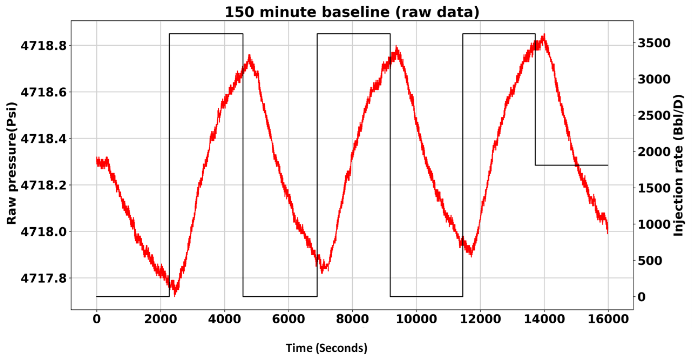

Figure 4. *Unprocessed/raw pressure data obtained from the pressure gauge installed at well F2. The rates are plotted on secondary axis. It can be observed from the image that the pressures exhibit a net upward trend due to continued injection. A detrending is required for pressure data set.*

Data processing

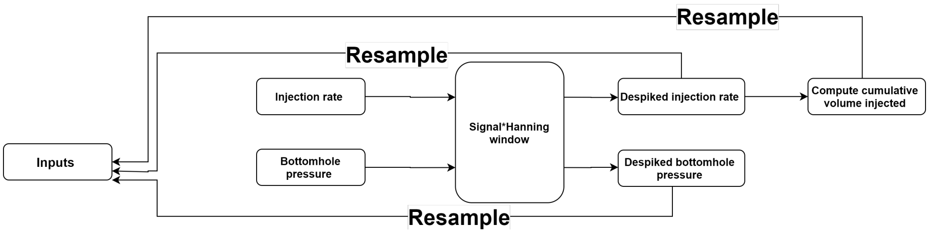

Figure 5. *Data processing workflow for this study. The data is de-spiked and re-sampled before modeling.*

Once we re-sampled the data, and both baseline and leak tests have the same sampling rates, we need to de-spike the data. The re-sampled data has 38,912 points. To de-spike the data, we used a Hann window of 21 samples and convolved the samples in the Hann window with the scaled data.

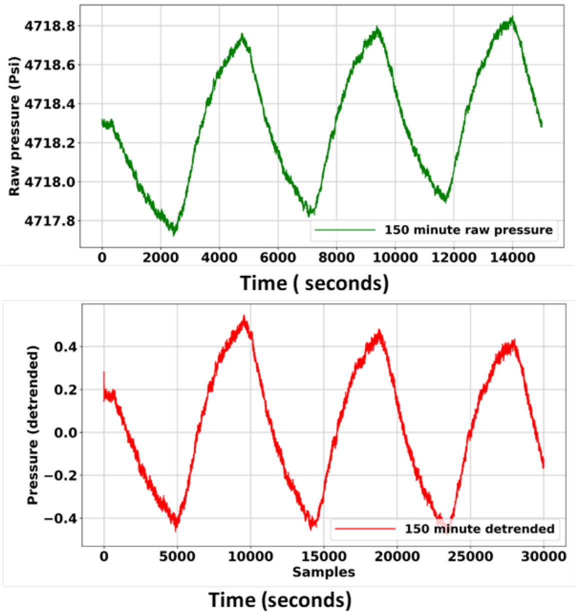

Figure 6. *150- minutes baseline test pressure response before(in green) and after(in red) detrending the data.*

A Hann function of length L (Harris, 1978) is defined as :

$$w_0(x) \overset{\triangle}{=} \begin{cases} \frac{1}{2}(1 + \ \cos(\frac{2\pi\, x}{L})) \ = \cos^2(\frac{\pi\, x}{L}), & |x| < \frac{L}{2} \\ 0, & |x| > \frac{L}{2} \end{cases} \tag{1}$$

This can then be re-sampled for a discrete signal as :

$$w(n) \ = \ w_0 \ \left\{ \ \frac{1}{2}[1 - \cos(\frac{2\pi\, n}{N})] \ = \ \sin^2(\frac{\pi\, n}{N}) \quad 0 \leq n \leq N \right. \tag{2}$$

where, N+1 is the length of the window.

For a signal, s(t) this window can then be convolved over the signal as:

$$S(t)* = s(t) * w(N+1) \tag{3}$$

where S(t)* is the smooth signal in the time domain, and H(N+1) is a Hann window of N+1 samples.

As injection rates remain the same for all tests, the rates alone cannot be used to predict the different pressures for the same rates. To overcome the issue, we created an additional feature of injected cumulative volume. This feature is then added as an input along with the injection rates to predict the given pressure response. Cumulative volume feature is nothing but the integral of the injection rates over a given period as:

$$C \ = \ \int_0^T q \ dt \tag{4}$$

where, C is the cumulative injected volume, q is the instantaneous rate and T is the time until injection is completed. We used de-spiked surface pump rates as the injection rates to compute the cumulative injection rate.

Figure 6 shows that there is a linear upward trend in the measured pressure in green and hence a detrending of the data is required. Without detrending, the machine learning models would have to extrapolate the input behavior, which is not ideal for anomaly detection tasks. The resampled and detrended data is shown in red. Notice that the number of samples are now higher and the linear upward trend is removed. As a final step for the input preparation, we used the min-max scalar from the Python 3.7 distribution Scikit-Learn library (Pedregosa et al., 2011) to rescale the inputs for modeling.

Figure 7 shows a comparison between the measured pressures in monitoring well F2 for the 90-minutes baseline and 90-minutes leak cases. The data in 7 shows that the pulse test is an effective method to distinguish between the leak and baseline scenarios and two pressure signals can be segregated. However, The absolute difference in pressure baseline and leak tests is minimal 0.5 Psi. Hence, the modeling techniques for prediction requires excellent accuracy.

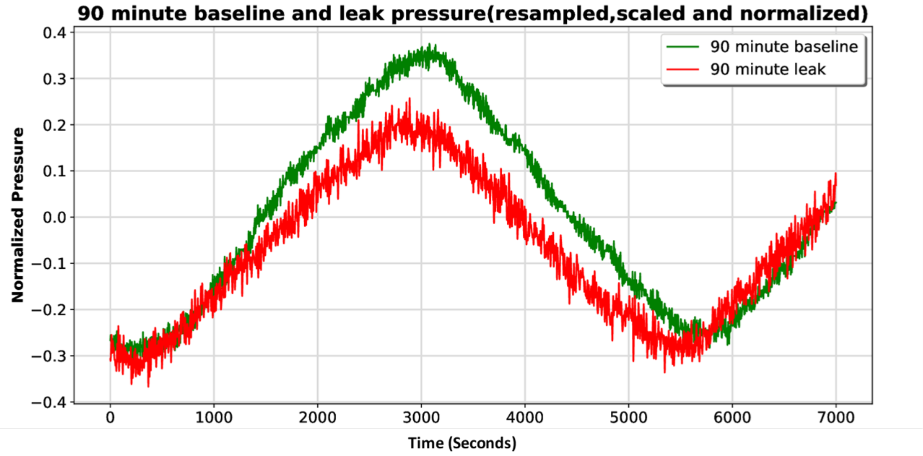

Figure 7. *Comparison of pressure response between the leak vs. non leak 90-minutes test obtained from pressure gauge installed at well F2. It can be observed that pulse test pressure response can distinguish between the leak vs. the non -leak case.*

Problem formulation as anomaly detection

Anomaly/outlier detection in a multivariate time series data is a well-defined problem in statistics and is investigated in depth by various authors e.g., Kandhari et al. (2009), Zimek and Filzmoser (2018) and, Wu et al. (2008). Simply put, an anomaly is an observation point for which the error exceeds a threshold for the given trained model. This concept can be extended from a single location to a sequence of observations. In this work, we defined anomaly using the squared error (ε) from the model predicted value to the actual observation. To further refine the analysis, we set thresholds on the (ε) to highlight the regions of anomaly. For all the analyses, the (ε) is defined as:

$$\varepsilon = (y_i - \widehat{y})^2 \tag{5}$$

where, y$_i$ is the measured value and $\widetilde{y}$ is the predicted value.

Neural Network Architectures

Modeling terminology used in the study

For all the deep learning architectures, we used a sliding window (Glumov et al., 1995; Vafaeipour et al., 2014) of 1000 samples to make the future prediction. Through trial and error, we found that a window of 1000 samples can sufficiently represent the features present in the data. Smaller windows are incapable of capturing the cyclical events properly, while larger windows leave subtle inflections in the pressure data.

The inputs for all models remain as: pressure measured at the monitoring well F2, injection rates obtained from injection well F1, and cumulative injection volumes which is computed from the injection rates at well F1. Hence, the input in all cases is a three-dimensional vector of 1000 samples each and the output is one or multiple pressure data points ahead in the future.

We used 150-minutes baseline and 90- minutes leak data for the analysis. The 150-minutes leak test was not included in this workflow. We implemented a total of 39,202 data points for analysis after re-sampling of the data. Out of these, 24,542 samples belong 150-minutes baseline test and 14,660 samples belong to 90-minutes leak test. For training purposes, we used 15,000 samples from 150-minutes baseline test to train the models — roughly 40% of the data. For the test set, we used the remainder of 9,452 samples from 150-minutes baseline and all 14660 samples from leak test. These data points are separated by various color arrows showing the train, test, baseline and leak boundaries on all result plots. The color coding of the data is explained in the results section.

The models are not trained on the leak test data, because the ultimate goal of our study is to determine the anomalous pressure behavior as compared to a baseline test. If the leak test was included in the analysis, the model could not distinguish between the baseline and leak test. Therefore,it would not be able to detect anomalous behavior of pressure data. In addition, we used mean squared error (MSE) as a loss metric in all models.

The chronological order of the tests shown in Table 1 shows that the 90-minutes leak test follows the 150-minute baseline test. As the reservoir pressure is a dynamic property and changes with injection, we used the 150-minute baseline and 90-minutes leak in the analysis.

The neural networks are intrinsically stochastic hence, we generated 20 realizations of each model to report the average training MSE and computational time. In all the architectures described below in the following sections we tested the model performance in terms of MSE and computational time to achieve best results in the least computational times. To avoid over-fitting, we used 20% dropout in hidden layers (Srivastava et al., 2014). Also. we attempted to forecast multiple points in the future in a multivariate and multistep fashion, to test the ability of a model to forecast more than one sample in the future. Figure 8 shows the models and their variations included in this study.

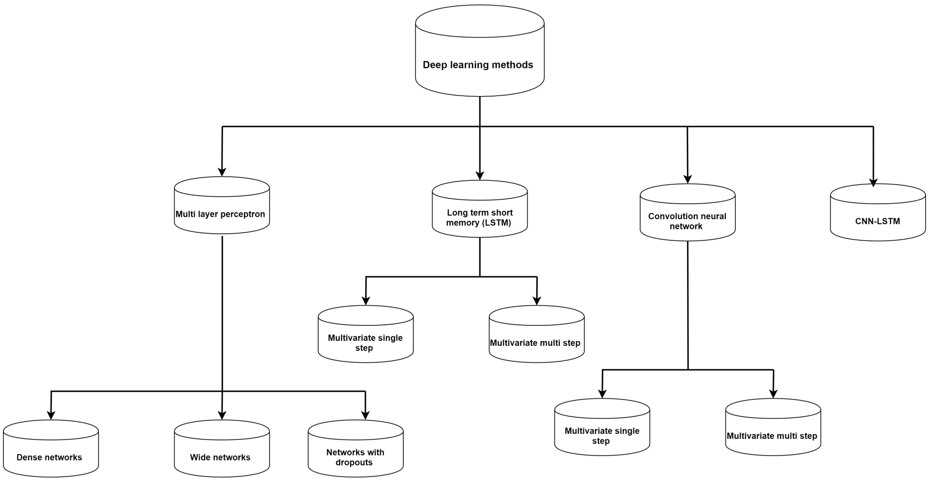

Figure 8. *Deep learning methods and their variants used in this study. A multistep forecast is added to the analysis to test the ability of the model to predict multiple samples in the future.*

Multilayer Feedforward Neural Network (MFNN)

An MFNN is a class of feed forward artificial neural network (ANN) which uses neurons as the basic computational unit (Yilmaz and Kaynar, 2011; Rynkiewicz, 2012; Amid and Mesri Gundoshmian, 2017). An artificial neuron is a computation unit consists of weighted inputs and outputs. The output signal is triggered by an activation function. One such simple computational unit is shown in Figure 9.

The weights on the inputs are analogous to the regression coefficients in simple linear regression. These are often initialize with random values. These weights are updated during the "learning process" of the model and are the desired model parameters.

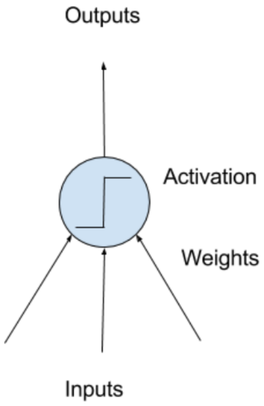

Figure 9. *A single computational unit i.e., a neuron. The inputs are biased by weights and the neuron is activated via an activation function to produce an output (Brownlee, 2018)*

.

An activation function/ transfer function governs the threshold after which the output signal is generated. We used the term activation function in the rest of this thesis. An array of activation functions have been developed over the years such as sigmoid, hyperbolic tangent, rectified linear unit (ReLU), leaky ReLU, Exponential Linear Unit (ELU) etc. Nwankpa et al. (2018) discussed these activation functions in detail.

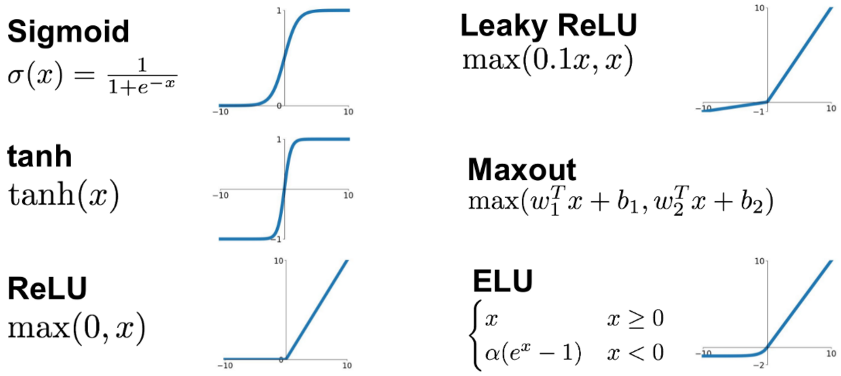

Figure 10. *Various activation functions commonly used in the deep learning architectures (Jia et al., 2014)*

.

Figure 10 show these activation functions and their activation behaviour. In this thesis, we used ReLU as the activation function.

A network of such artificial neurons form a layer and modern deep learning architectures use multiple layers of these neurons to form extremely, complex and dense multi-layer models. One such network architecture is MFNN. The network must have an input layer that receives the inputs. One or multiple hidden layers and an ouput layer. Output layer is the layer which produces the desired prediction. The network learns by using backpropagation algorithm Mozer (1995); Sathyanarayana (2014) based on some loss function metric e.g., MSE. The learning of the network is optimized by an array of available optimizers such as stochastic gradient descent (SGD), RMSprop, Adaptive gradient (Adagrad), Adaptive moment tensor (ADAM) etc.

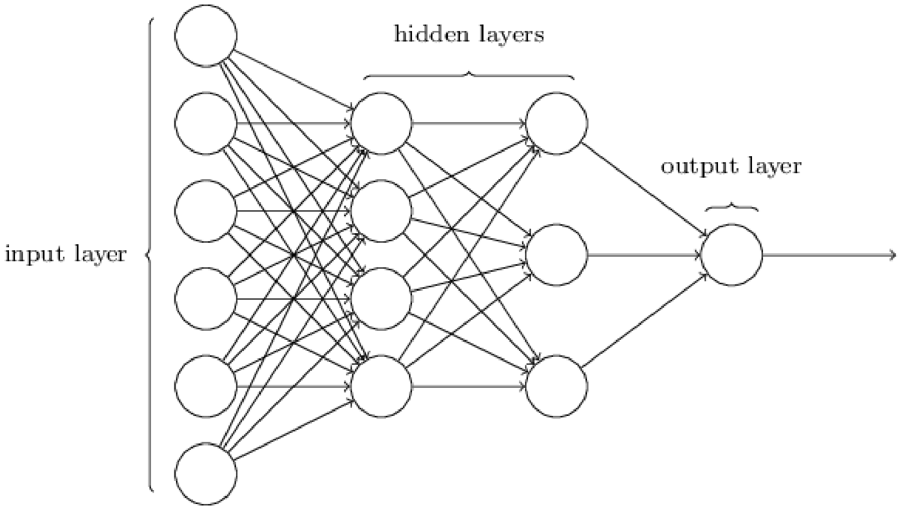

Figure 11. *Typical MFNN architecture. A basic MFNN architecture has an input layer, a set of hidden layers and an output layer.*

After testing both the number of layers and the number of neurons per layer, we selected a MFNN model composed of three hidden layers, respectively with 20, 20, and 10 neurons with Rectified Linear Unit (ReLU) as an activation function(Behnke, 2003). The final output is the pressure at the next time step, and hence the last layer has just one output. We used adaptive moment estimation (ADAM) (Kingma and Ba, 2014) with a learning rate of 1e-04.

To extend this idea of predicting one point forward in time, we attempted to forecast multiple points in the future to see the ability of the model to predict multiple points before the window is slid to the next set of input data. This technique is known as multivariate, multi-point prediction. The network architecture is similar to the one in multivariate single point described above. However, there are essential differences in the data preparation of both these methodologies. We used the 1000 samples to predict the first one, then ten, and finally 1000 samples in the future. The results of the model are summarized in section three. The prediction is not included in the input data to predict the future samples but the input is divided into 1000 samples window and then 1, 10, 100 and 1000 samples are used for prediction forward.

Long Short-Term Memory (LSTM)

LSTM falls under recurrent neural network architecture. Recurrent neural networks (RNN) intrinsically have a feedback loop allowing for memory retention in longer sequences of data LeCun et al. (1995); Hochreiter et al. (2001); Nguyen et al. (2019). RNN family of network architectures thus, are naturally suited for time series prediction. LSTM are trained via backpropagation in time to avoid vanishing gradients which a typical RNN structure suffers from.

The network consists of memory blocks instead of a single neuron which connect into sequential layers to make dense layers. The memory block is ”gated” where each gate has its own activation functions. The gates decide which information to keep or discard. A typical block has three gates, namely, input gate, output gate and the forget gate. Figure 12 shows the conceptual layout of such memory gate.

The forget gate decides which information to keep/discard. The input gate updates the memory state and the output gate decides the output of that memory block. A series of gated memory blocks/cells allows the network to read sequences of data and not just single point.

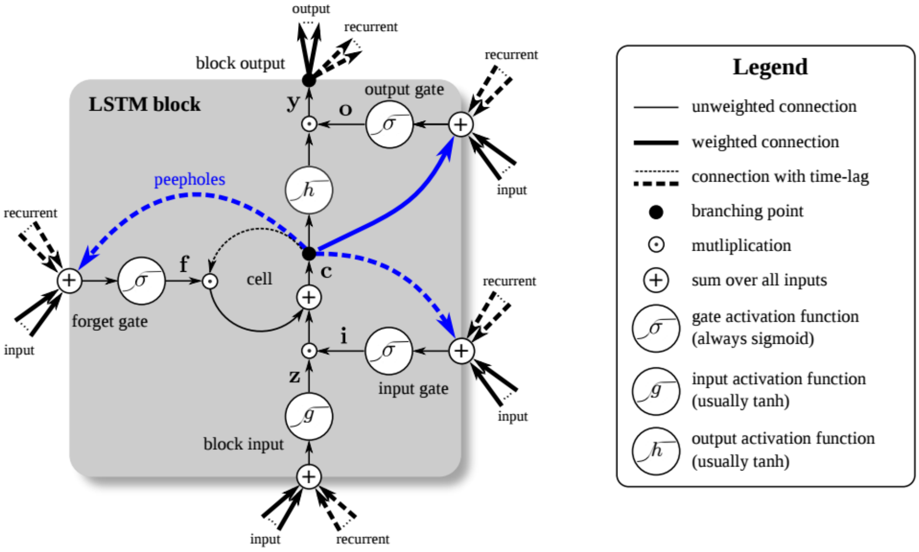

Figure 12. *Long Short-Term Memory (LSTM) unit. The LSTM unit has four input weights (from the data to the input and three gates) and four recurrent weights (from the output to the input and the three gates). Peepholes are extra connections between the memory cell and the gates, but they do not increase the performance by much and are often omitted for simplicity (Greff et al., 2017).*

LSTM is a state-of-the-art class of RNN which are highly effective in predicting the future response based on the past training history of some of the most complicated tasks such as language translation (Volkova et al., 2013). Stacked with multiple layers of LSTM they become even more powerful (Fernández et al., 2007). In our case, a simplistic LSTM has proven to be sufficient to characterize our problem.

To train our model, we employed a series of 100 such units in a densely connected network with RELU as the activation function and ADAM as the optimizer, as well as a learning rate of 1e-04. Furthermore, to determine the model scalability we followed a similar methodology as in the MFNN architecture, using a window of 1000 samples to forecast one, ten, 100 and then 1000 samples.

Convolutional Neural Networks (CNN)

A CNN architecture commonly consists of three set of layers: convolutional layers, pooling layers, and fully connected layers. A convolutional layer convolves the input with a series of filters to produce a filtered output. The filtered result then is commonly used as input to a pooling layer which reduces the dimensionality of the data. The output from the pooling layer is flattened and passed to a fully connected dense layer or a series of layers Khan et al. (2019); Zeiler and Fergus (2014); Simonyan and Zisserman (2014); Gu et al. (2018). The neurons in a CNN architecture, unlike MFNN, may not receive the inputs from all previous neurons but only from a selected "patch" from the previous layer hence reducing the chance of overfitting as compared

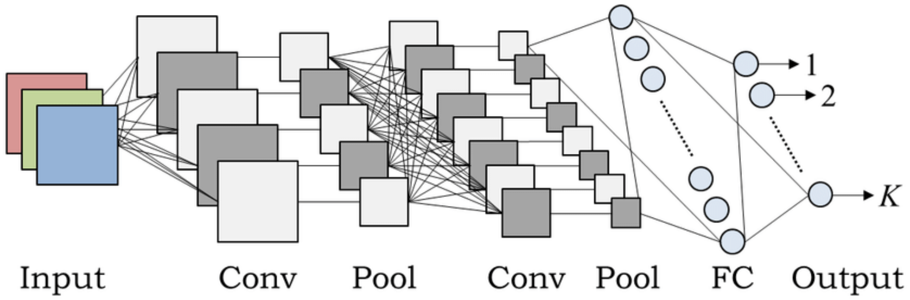

Figure 13. *Major components of a CNN architecture (Hidaka and Kurita, 2017)*

to the MFNN. Figure 13 shows a conceptual implementation of the CNN.

The CNN model consists of a single convolutional layer with filter of size three, and RELU activation function, a max pooling layer of size two and stride two, followed by two fully connected layers comprising 50 neurons and RELU activation function. The output of the network is a single neuron that contains the pressure at a future time step. We included ADAM as the optimizer with a learning rate of 1e-04.

Convolutional- LSTM (CONV- LSTM)

As discussed earlier, the merit of LSTM lies in keeping the temporal aspect of sequence data. CNN on the other hand is a powerful architecture for extracting features in both image and sequence data. Therefore, we selected a hybrid version of CNN and LSTM for the pressure forecasting. CNN was implemented to extract complex shapes from the time series data, and LSTM to keep track of these features over longer sequences.

Hybrid models are very efficient in fulfilling both tasks (Xingjian et al., 2015a).

We divided the 1000 samples window of three features into ten subsets of 100 samples, each which is fed first to the CNN for feature extraction. CNN then forwards this information to LSTM for temporal updates. To achieve this task, we used time distributed layers in the architecture (Xingjian et al., 2015b). The CONV-LSTM model is composed of one convolutional layer with 64 filters with size three and RELU activation function, followed by a max-pooling layer, the output then is flattened and passed as input to a LSTM layer with 50 units that outputs a sequence. The final LSTM sequence is then used as input to the output layer for the prediction of the pressure data.

Results

Evaluation metrics and terminology

This section discusses the results from the various network architectures described previously. Hereinafter, the terminology used for multistep varies according to the number of samples predicted in the future. For example MFNN_100 represents the case where 1000 samples fixed window is used to predict 100 samples in the future. The results show both 150-minutes baseline and 90-minutes leak data.

In a real life scenario, the models can be used for detecting anomalies on live data streams. In this case study, a real pressure behaviour with baseline and leak scenario is available. However, we need a model architecture to forecast a baseline pressure in the future to compare it with the recorded future pressures. Hence, we used a multi-step forecast. A model which can forecast 1000 samples in the future with confidence has a better scalability than a model that can only forecast 100 samples in future.

We used three diagnostic plots in the model architectures. In first diagnostic plot we compared the actual versus the measured value of the pressure. The second plot consists of the training process and the corresponding MSE after every training epoch, with a total of 100 epochs for training. The third plot is the key output of the study which is point wise squared error also referred as anomaly plot throughout this section. We included different errors in the diagnostic plots for the model. To generate the anomaly plot, we computed the squared error for every sample as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widetilde{y})^2 \qquad (6)$$

where MSE is the mean squared error, y$_i$ is the actual pressure measurement and $\widehat{y}$ is the predicted pressure. Higher error indicates anomaly. We filtered this error by setting all errors less than 10e-03 to zero and amplifying the errors higher than 10e-03 by a factor of 100. This is referred to as filtered error in the diagnostic plots. As smaller errors are suppressed and higher errors are amplified, it is easier for human interpreter to identify the anomalous pressure behaviour. The other error metric used is the mean squared error i.e., MSE. This is a metric to evaluate the machine learning models. Higher training MSE values imply a less accurate model during training. We made two different types of comparisons to evaluate the models:

1. Models which are scalable i.e., models which can forecast higher number of samples with same amount of training data. These are referred by a subscript in all the relevant Figures. For example $MFNN100$ refers to a multi-layer feed forward network where 1000 samples window is used to predict 100 samples in the future. If a fixed number of input samples are used to forecast a higher number of samples in the future, the model is expected to have higher errors. But, a model which gives unusually high errors as the number of samples in the future increases, is unstable and hence, undesirable. We used the log-loss (MSE) as a metric to evaluate if the

models are scalable, and we repeated the multi-step forecasting process for 1, 10, 100 and 1000 samples for all the model architectures.

2. Models which are the best, overall. For this criterion, we compared different model architectures. The comparison includes various observations such as : can a model sufficiently capture the sinusoidal pressure behaviour and not raise the false alarms during rising or falling pressure cycles? Can the model be trained in a reasonable time with technology available today? can the model take advantage of hardware such as graphics processing unit (GPU)?

In all the Figures, the first 15,000 samples corresponding to baseline tests are used for training and rest of the samples are used for prediction. The model training ends at 15,000 samples which is highlighted by a green arrow in the figures. Baseline test ends at 24,542 samples which is highlighted by a brown arrow. Table 2 summarizes the average results from 20 realizations of each model.

MFNN results

Figure 14 through Figure 16 show the results from the MFNN architecture. From Figure 14 it can be noticed that the model performs exceptionally well for both training and test set and predicted pressures are very close to the actual pressure values.

As the model is trained on baseline, it expects a baseline equivalent response

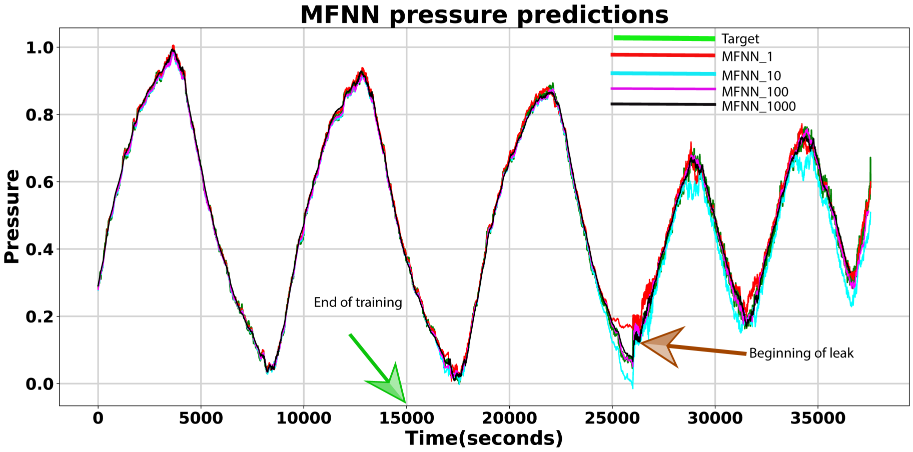

Figure 14. *MFNN results for baseline pressure and the predictions for the different models. Target pressure is shown in color green while the model outputs are shown in various colors. Green and brown arrows show the end of training and end of baseline test respectively. It is observed that the model produces excellent results for the pressure prediction for only 15,000 samples.*

everywhere and hence can detect the anomaly in the leak portion of the data. This is further highlighted in Figure 16 where the filtered MSE response shows the difference in predicted and actual values. We set all MSE below 10 e-3 to zero, and we amplified the MSE above 10 e-3 by a factor of 100. It can be observed from the filtered MSE in Figure 16 that except one stem of the data, the leak and no leak portions of the data can be clearly be identified as an anomaly or not.

The model is unable to capture some of the trough portions of the pressure waveform and unpredictable towards the last pumping cycle of 150 -minutes baseline

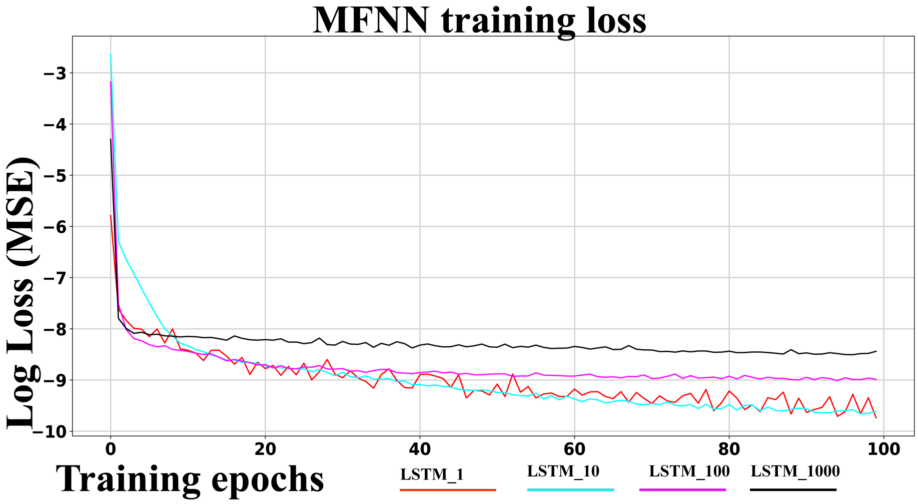

Figure 15. *Log-Loss (MSE) with number of trained epochs for all the cases, It can be observed that as the number of predictions in the future increases, both 1 step, ten steps and 100 steps yield consistent results with losses gradually increasing towards higher number of samples forecasted.*

test. It is essential to mention that in the final segment of the 150-minutes baseline is a pressure fall off, the surface pumps are shutting down, and hence the behavior is expected to be abnormal. Usually, with pump shut-downs, there is venting at the surface, which is essentially an induced leak.

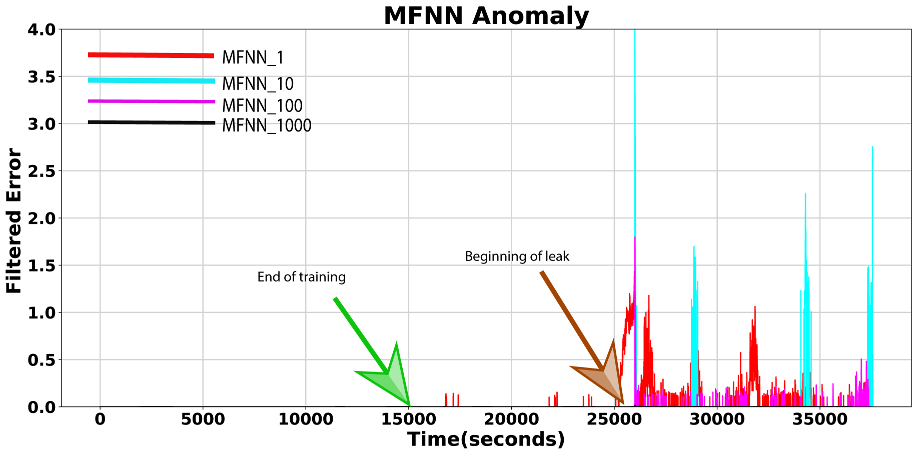

Figure 16. *Filtered error versus the number of samples. All errors less than 0.001 are set to zero and errors greater than 0.001 is amplified by a factor of 100. The end of training is shown by a green arrow and leak portion is shown by a brown arrow on x-axis. It can be observed that the MFNN can separate the leak and no-leak portions of the data.*

LSTM results

Figure 17 through Figure 19 shows that LSTM scales better than MFNN when same window size is used to predict various points forward in time. Figure 17 shows the pressure prediction from the LSTM architecture and Figure 18 shows the losses for different prediction interval cases. The MSE increases gradually as more points are added but there is a clear separation between the baseline and leak data.

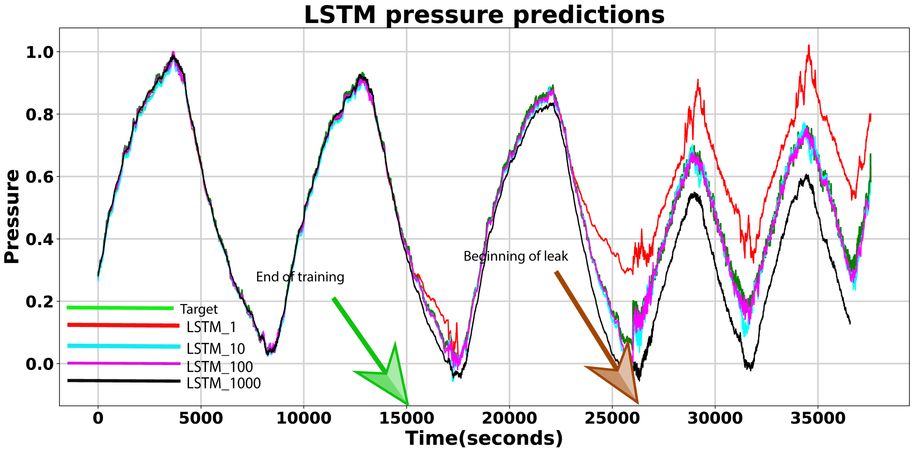

Figure 17. *LSTM pressure prediction. The target pressure is shown in color green and the predictions forward are shown in various colors in the legend. The end of training is shown by a green arrow while the start of leak test is shown by a yellow arrow.*

For anomaly detection, it is of utmost importance that the data are segregated instead of lower MSE values. Notice in Figure 19 , which is the LSTM anomaly the drastic differences between the baseline and the leak stems of the data. It is much

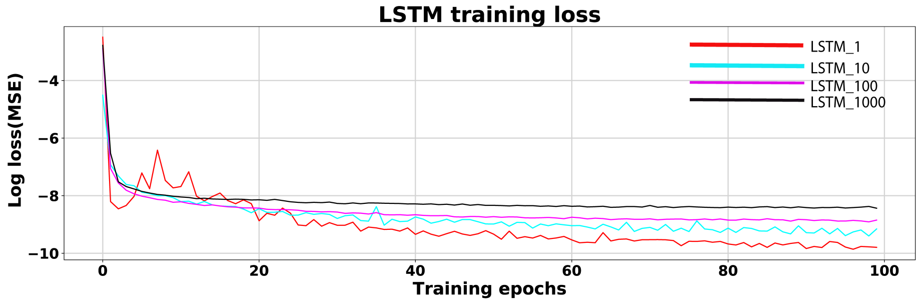

Figure 18. *LSTM log losses. It is observed that as the prediction interval increases, the net losses increase.*

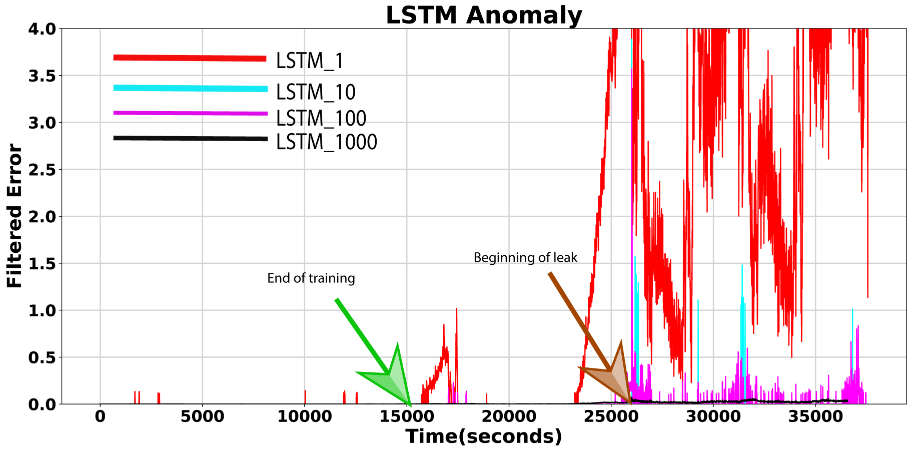

Figure 19. *LSTM anomaly. We observed that the LSTM can efficiently separate the leak and non-leak data. Notice the difference in results between the MFNN and the LSTM anomaly. The anomaly is more pronounced in the case of LSTM than the MFNN anomaly.*

easier to identify the anomaly in case of LSTM as compared to MFNN. Hence, for anomaly detection, LSTM is a preferred choice over MFNN in our case.

CNN results

Figure 20 through Figure **??** show the results from the training and the test data from the CNN architecture. Results in Figure 20, show that the CNN architecture can capture the sinusoidal wave forms efficiently. Figure 22 show that the overall MSE is higher in case of CNN as compared to LSTM, however in cases where smaller number of samples are used for future prediction, the model can distinguish between the leak and the baseline portions efficiently. CNN can also be parallelized on a graphical processing unit(GPU) and hence provide a computational edge over LSTM architecture.

It is evident that LSTM provides overall better results than CNN however CNN is not just computationally more efficient on a GPU device but also has he ability to capture the sinusoidal waveforms efficiently.

CONV-LSTM results

The hybrid architecture utilizes the features of both convolutional layer and the convolutional layer can be parallized on a GPU. Hence, state-of-the-art results can be obtained faster without training for larger epochs like recurrent network/LSTM sequential updates. We showed the results obtained from this architecture from Figure 23 to Figure 25. From Figure 23 we observed that this hybrid architecture is able to capture the sinusoidal features of the pressure signal efficiently. Figure 24 shows

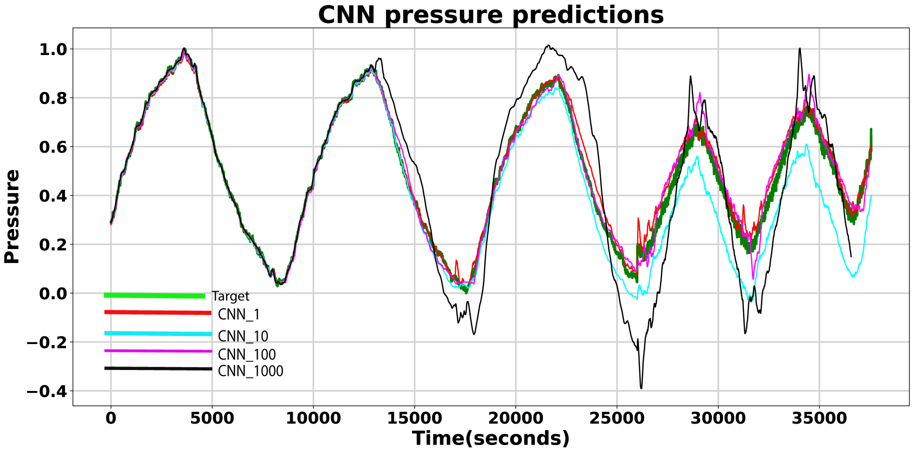

Figure 20. *CNN pressure prediction. The target pressure is shown in green and the predictions are shown in different colors. The end of baseline test is shown by a green arrow. The start of the leak test is shown by a yellow arrow. Except for 1000 samples prediction, all the other cases can capture the sinusoidal waveform efficiently. The green and yellow arrows show the end of training data and beginning of leakage data respectively.*

the log-loss MSE for this architecture with training epochs.

Notice from Figure 24, although similar epochs compared to any other architecture, CONV-LSTM takes the highest amount of time for training. However, the log loss shows that the algorithm converges rapidly thus eliminating the need for training for higher number of epochs, giving it an advantage over other architectures. The filtered MSE anomaly shows that the architecture can differentiate between the baseline and the leak tests.

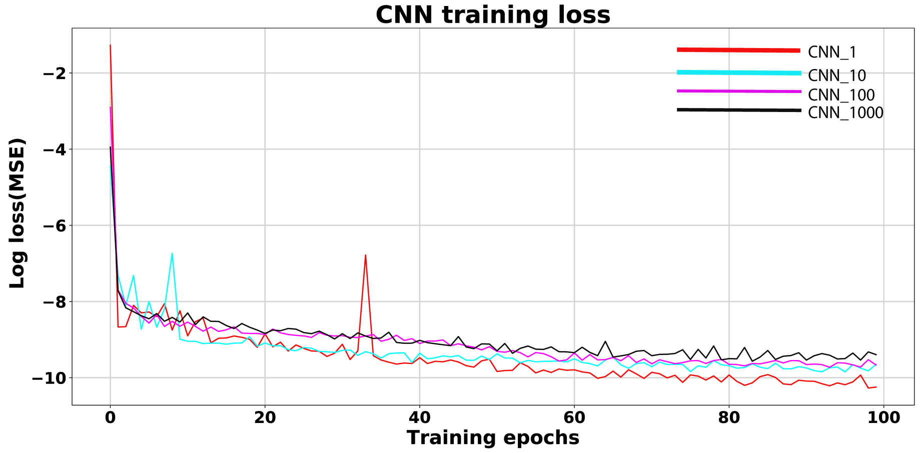

Figure 21. *CNN training log-losses. As the number of samples foretasted in future increases, the training losses increase. For smaller number of predicted samples in the future, the CNN results are comparable to that of LSTM network.*

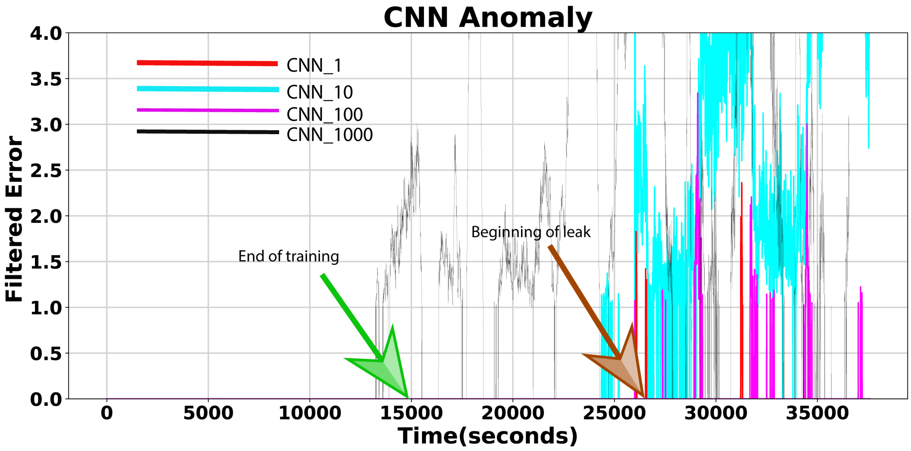

Figure 22. *MSE CNN anomaly. The CNN can effectively distinguish the leak and the non leak portions of the data except the CNN 1000 case (shown in light grey color). The end of training data is shown by a green arrow and start of leak is shown by the yellow arrow. Notice the anomaly spikes after the yellow arrow which is the data corresponding to 90-minutes leak.*

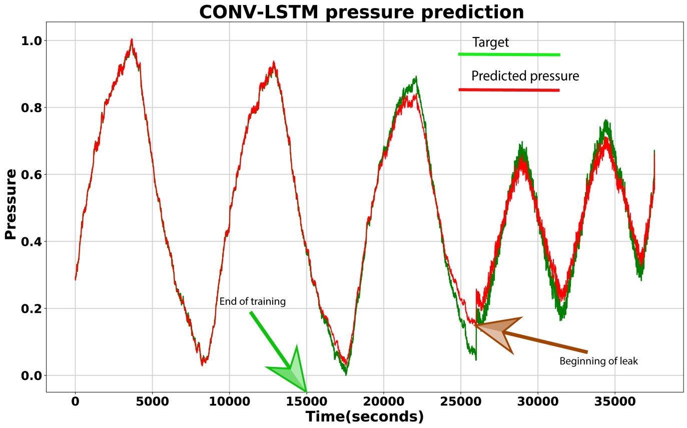

Figure 23. *CONV-LSTM pressure predictions. The CONV-LSTM results are comparable to LSTM. The architecture is able to capture the subtle features of the sinusoidal pressure signal.*

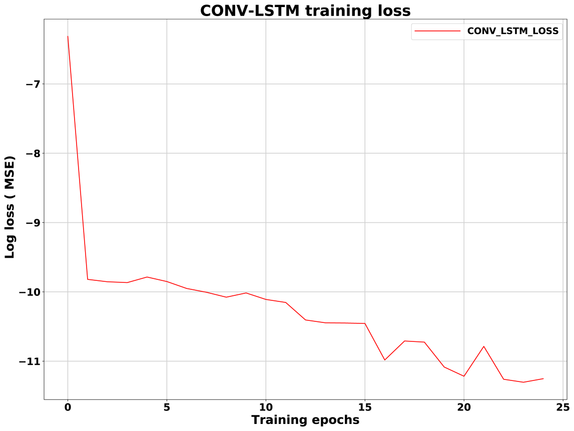

Figure 24. *Log loss (MSE) for the CONV- LSTM architecture. It is observed that the model converges rapidly with the least training MSE of all architectures discussed before.*

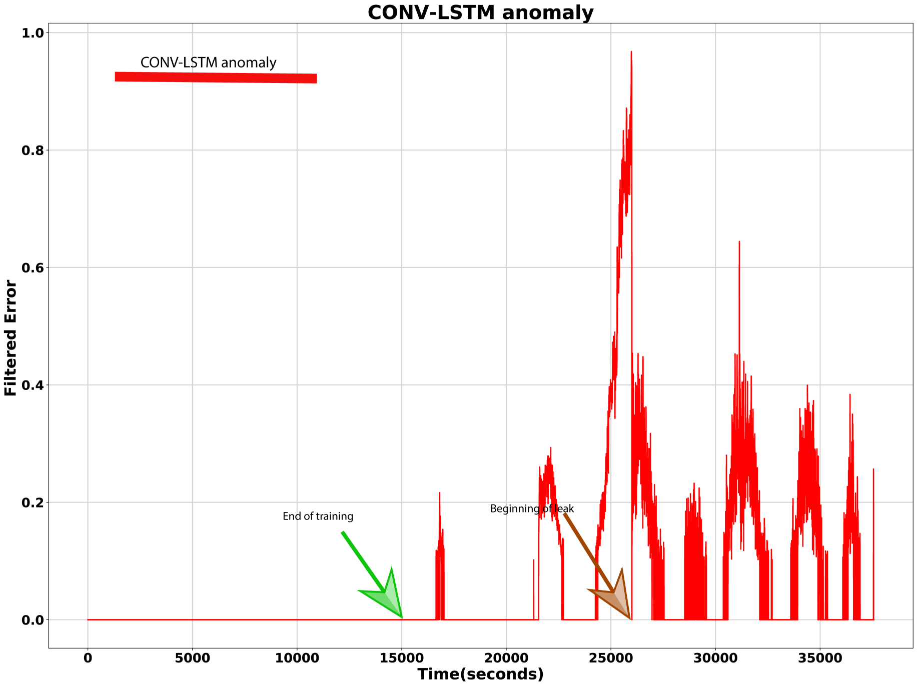

Figure 25. *The CNN-LSTM prediction and the MSE. It can be observed that the model can distinguish between the leak and non-leak portions of the data efficiently.*

Model	**Training MSE**	**Training epochs**	**Processor time (seconds)**
MFNN_1	5.8 e-05	100	82
MFNN_10	6.6 e-05	100	83
MFNN_100	1.2 e-04	100	81
MFNN_1000	2.1 e-04	100	80
LSTM_1	5.5 e-05	100	328
LSTM_10	1.01 e-05	100	332
LSTM_100	1.4 e-05	100	334
LSTM_1000	2.1 e-05	100	328
CNN_1	3.5 e-05	100	226
CNN_10	6.4 5e-05	100	213
CNN_100	6.3 e-05	100	209
CNN_1000	8.3e-05	100	193
CONV-LSTM_1	1.2e-06	25	3800

Table 2. *Results summary for all models used in the study.*

Model extension to different scenarios

We discussed earlier in the manuscript, the dynamic nature of the hydrocarbon reservoir under CO_2 injection. As the injection continues, the average reservoir pressure increases, and the stresses across the sealing faults change. In addition to rock properties such as porosity and permeability, that can change as function of the stress regime as well.

Hence, a natural question arises: can our workflow be possibly borrowed and applied to any other pulse test? To answer this question, we included another set of data. This data set consists of: 90-minutes baseline test, 150-minutes baseline test and 90-minutes leak test. Before in all our analysis we have only utilized 150-minutes baseline test and 90-minutes leak test.

This is shown in Figure 26. We used first 10,000 samples from the 90-minutes baseline test to train a MFNN model and is shown by the black arrow. The end of 90-minutes baseline test or start of 150-minutes baseline test is shown by a green arrow. The start of 90-minutes leak test is shown by a yellow arrow. This convention is held throughout this section. Using the trained model from first 10,000 samples belonging to 90- minutes baseline, we predicted the pressure behavior for both 150-minutes baseline test and the 90-minutes leak test. The results are summarized in Figure 26 and Figure 27.

We noticed that when the model is trained only on 90-minutes baseline

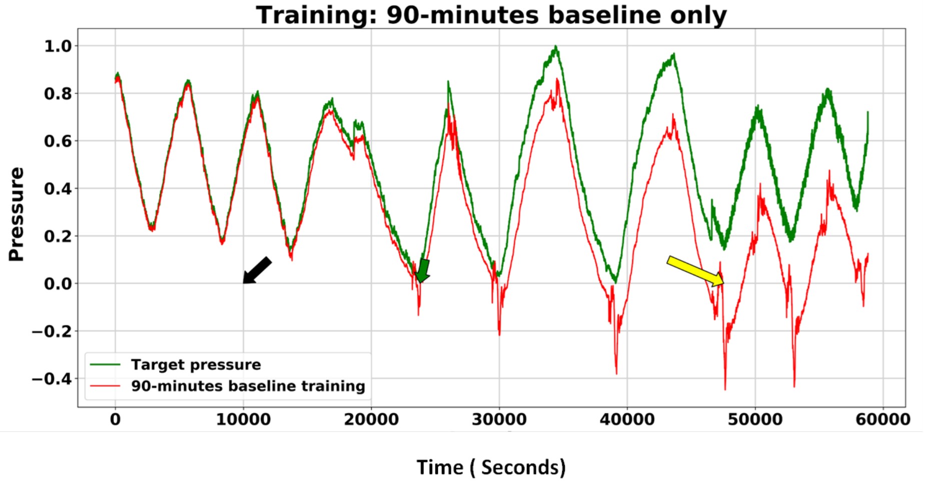

Figure 26. *Pressure prediction using MFNN for: 90-minutes baseline + 150-minutes baseline + 90-minutes leak test. 10,000 samples from 90-minutes baseline test is used for training the MFNN and rest of the samples are used for prediction. The end of pressure waveform used for training is shown by a black arrow. The end of 90- minutes baseline test is shown by a green arrow and the start of the 90-minutes leak test is shown by a yellow arrow. Notice that the model prediction is worse in case of both 150-minutes baseline and 90-minutes leak tests.*

test, it identifies the 150-minutes baseline as well as 90-minutes leak test, both as an anomaly. MSE is higher in case of 90-minutes leak test than 150-minutes baseline test but both are much greater than the 90-minutes baseline test which the model is trained on. Hence, as expected a model can only learn from the training data distribution and will address everything else as an anomaly.

Now, a second question arises, can the model learn from a new data set

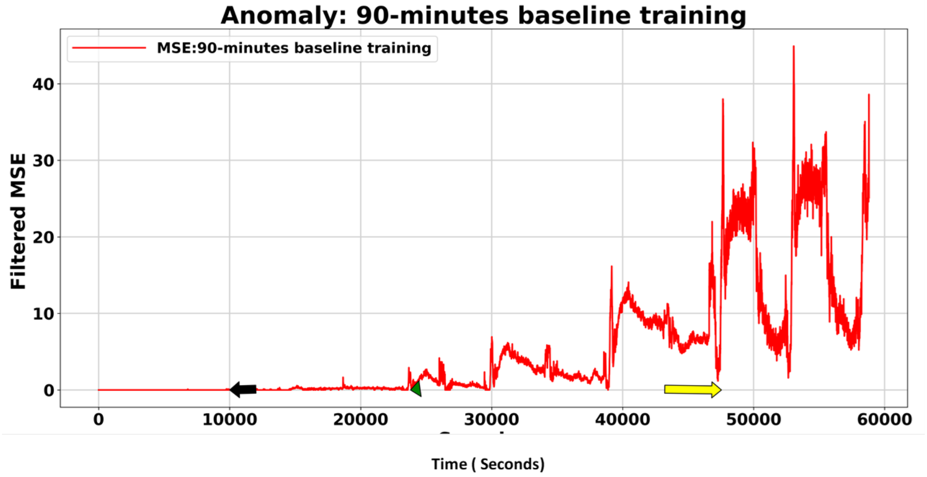

Figure 27. *Filtered MSE for the model when trained only on 90-minutes baseline data set. All MSE below 10e-03 are set to zero and all MSE above 10e-03 are amplified by a factor of 10e+02. Notice the beginning of the 150-minutes baseline test shown by the green arrow. The high MSE between the green arrow and the yellow arrow (start of 90-minutes leak) indicates an anomaly even in 150-minutes baseline case i.e. a leak. Hence, the model fails to distinguish between the 150-minutes baseline and 90-minutes leak tests. The 90-minutes baseline test is still classified as no-leak.*

and incorporate the additional features and reservoir behavior at different states? To answer this question, we included only one-half cycle of the 150-minute baseline test into our training in addition to 10,000 samples from 90-minutes baseline test. This is shown in Figure 28. The black arrow shows the 10,000 samples used for training from 90-minutes baseline test. The two cyan arrows show the section of the 150-

minutes test included in training. The green and yellow arrows show the beginning of 150-minutes baseline test and 90-minutes leak test respectively.

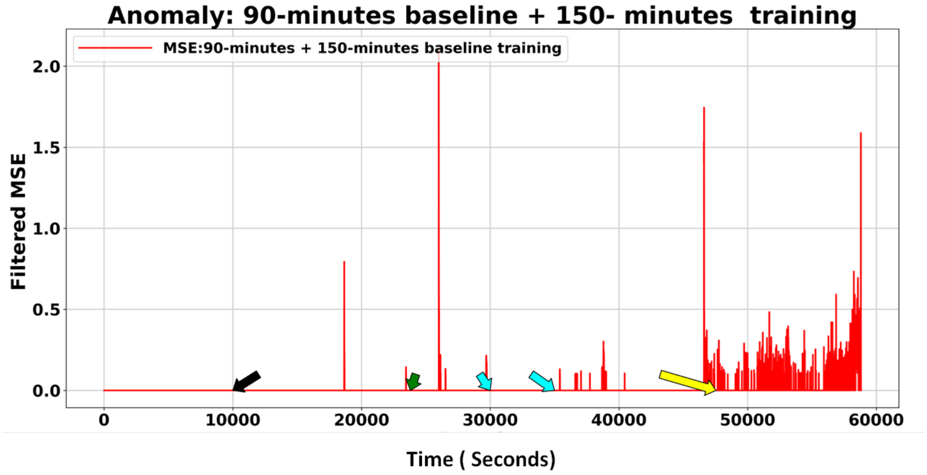

Figure 28. *Pressure prediction for 90-minutes baseline, 150-minutes baseline and 90-minutes leak tests. The cyan arrows show the data utilized from 150-minutes baseline test in the training process in addition to the 10,00. samples from 90-minutes baseline test. Notice that even 5,000 samples increase the pressure predictions dramatically.*

Figure 28 and Figure 29 show the MFNN results on the new training data. It can be observed from the results that the model can now easily distinguish between the baseline and leak tests effectively. Thus, even if a small portion of a different data set is included in the training, the model is powerful enough to learn new features. In a nutshell, deep learning architectures are powerful enough to learn new features quickly; however, as expected they can only make a prediction based on the data distribution of the data set used for training. Deep learning models lack intuition, unlike human interpreters. Hence, for full-scale field deployment, the models have to be trained on a particular set of test parameters.

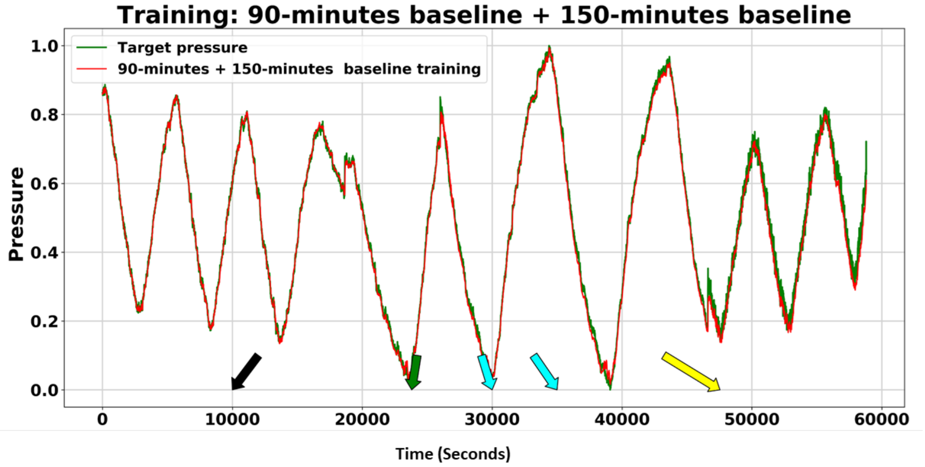

Figure 29. *Filtered MSE for the case in which 5,000 samples from the 150-minutes baseline test is included in the analysis. The portion of data included from the 150-minutes baseline test is shown by cyan arrows. Notice that the 90-minutes leak test is now clearly distinguished from the baseline test as compared to Figure 17 and highlighted as an anomaly.*

Based on the model's capability on multi-step forecasting, a model can be used for future prediction. As the system is dynamic, once a forecast with the high degree of confidence interval has already occurred, this data must be included in the training, and the model has to be updated. As our workflow utilizes time-series data, the additional training is fast and efficient, making the model practically scalable and field-deployable.

Novelty of the Study

This work automates the process of leak detection in CCS projects using deep learning on time series data which is computationally efficient than processing two dimensional images or three dimensional voxels. We employed this methodology on experimental data obtained from Cranfield reservoir, Mississippi, USA. However, the method can be extended to live pressure feeds recorded continuously from the wells when the reservoir is on planned injection schedule.

The multi step forecast in my models show promise in forecasting multiple data points in the future and hence can be used as a proxy for forecasting instead of reservoir simulation. These forecasts then can be included in training data and model can be updated in time. This eliminates the need for multiple physics based simulation forecasts and reduce the computational cost.

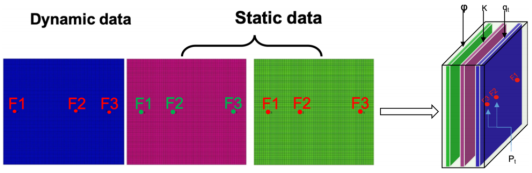

Figure 30. *Input data preparation for the CONV-LSTM study. The data consists of both static and dynamic data for reservoir characterization (Zhong et al., 2019a)*

Figure 30 and Figure 31 shows the input data and the workflow adopted

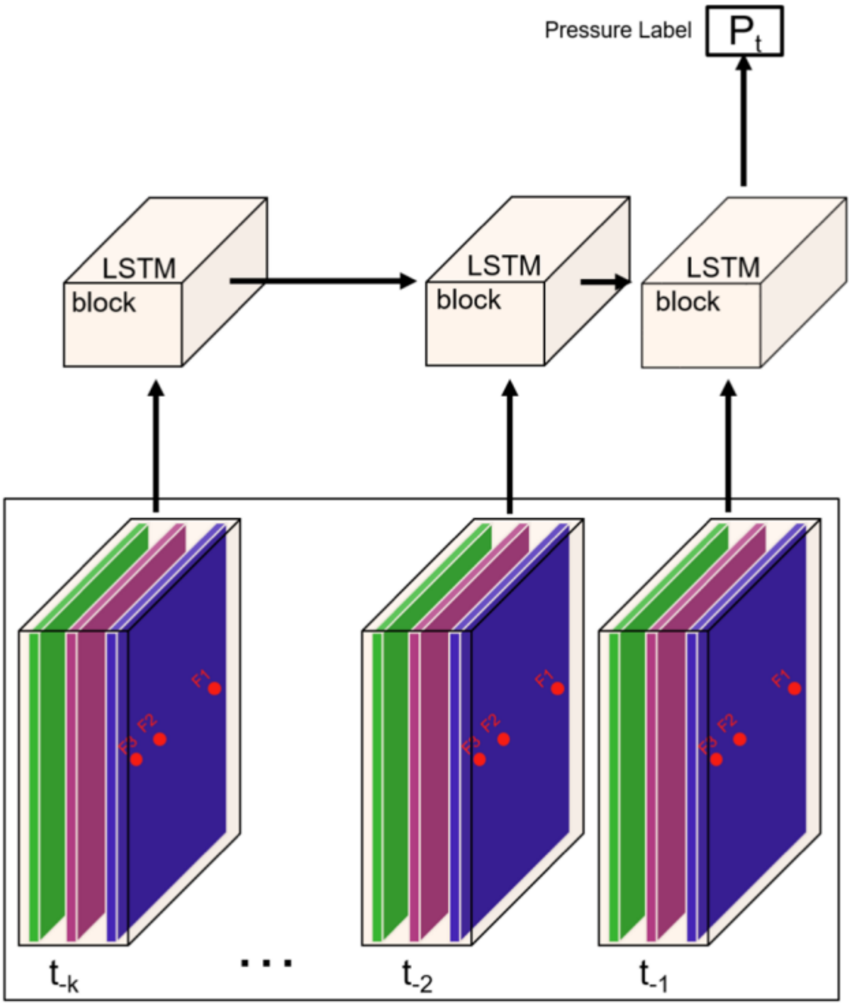

Figure 31. *Workflow adopted by Zhong et al. (2019a). Notice that the CONV-LSTM requires spatio-temporal analysis of two-dimensional images.*

by Zhong et al. (2019a). The architecture used in their study is CONV-LSTM but it utilizes two dimensional images obtained from a physics based tool such as a reservoir simulator. Prior to simulation a through reservoir characterization is necessary in their proposed workflow. The results are compared once a model is trained with the actual simulator and differences in the pressure responses between baseline and leak cases is used as an leak detection. The model also is very sensitive to the number of layers used in the modeling as well as permeability of the reservoir.Training these models are computationally inefficient. The field conditions such as injection rates change dynamically which will ultimately requires to re-train the models.

Another such study was conducted by Sun et al. (2014b). The authors show that by analyzing the pressure response of the baseline versus the leak well in frequency domain the two responses can be distinguished. This methodology requires computing the DTFT from the pressure response continuously and then analyzing the results. The automation of the spectral responses will require not just computing the DTFT but also using different machine learning architectures to distinguish them on two dimensional images thus adding computational cost.

Conclusions

In this work, we discussed the applications of deep learning architectures for leakage detection in carbon sequestration projects. In our findings, MFNN architecture performs excellent for training and test data sets but struggles to clearly distinguish between baseline and leak test as prediction in time interval increases. When it comes to predicting the future response, the LSTM architecture performs the best in our results overall, but the convolutional neural network (CNN) makes an excellent case for capturing the sinusoidal waveform like features in our dataset. CNN also can be parallelized on a GPU device, giving a computational advantage over LSTM which is processed sequentially. The hybrid architecture of CNN and LSTM, i.e., CONV-LSTM, provides the best of both worlds in terms of leak identification and computational times. However, computational times can be a limitation on this architecture. All machine learning algorithms, including deep learning architectures, learn from training data, and CCS are a dynamic system affected by complex geology and operational parameters. However, we showed that deep learning architectures can be extremely useful for an automated early warning system for leak detection in CCS projects which can then be investigated with conventional engineering methods by human interpreters.

Bibliography

Amid, S., and T. Mesri Gundoshmian, 2017, Prediction of output energies for broiler production using linear regression, ANN (MLP, RBF), and ANFIS models: Environmental Progress & Sustainable Energy, **36**, 577–585.

Bachu, S., 2000, Sequestration of CO_2 in geological media: criteria and approach for site selection in response to climate change: Energy conversion and management, **41**, 953–970.

———, 2003, Screening and ranking of sedimentary basins for sequestration of CO_2 in geological media in response to climate change: Environmental Geology, **44**, 277–289.

Behnke, S., 2003, Hierarchical neural networks for image interpretation: Springer, **2766**.

Bergmann, P., U. Lengler, C. Schmidt-Hattenberger, R. Giese, and B. Norden, 2010, Modelling the geoelectric and seismic reservoir response caused by carbon dioxide injection based on multiphase flow simulation: Results from the CO_2 SINK project:

Geochemistry, **70**, 173–183.

Brigham, W., et al., 1970, Planning and analysis of pulse-tests: Journal of Petroleum Technology, **22**, 618–624.

Brownlee, J., 2018, Deep learning for time series forecasting: Predict the future with MLPs, CNNs and LSTMs in Python: Machine Learning Mastery.

Castelletto, N., G. Gambolati, and P. Teatini, 2013, Geological CO_2 sequestration in multi-compartment reservoirs: Geomechanical challenges: Journal of Geophysical Research: Solid Earth, **118**, 2417–2428.

Chadwick, R., R. Arts, M. Bentham, O. Eiken, S. Holloway, G. Kirby, J. Pearce, J. Williamson, and P. Zweigel, 2009, Review of monitoring issues and technologies associated with the long-term underground storage of carbon dioxide: Geological Society, London, Special Publications, **313**, 257–275.

de la Iglesia, A. G., A. Malehmir, B. Lund, C. Juhlin, and R. Carbonell, 2018, High-resolution seismic imaging of the garpenberg ore system (central sweden) through active-and passive-seismic survey: 2nd Conference on Geophysics for Mineral Exploration and Mining, European Association of Geoscientists & Engineers, 1–5.

Dondurur, D., 2018, Acquisition and processing of marine seismic data: Elsevier.

Fernández, S., A. Graves, and J. Schmidhuber, 2007, Sequence labelling in structured domains with hierarchical recurrent neural networks: Presented at the Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI 2007.

Fernández-Montiel, I., M. Touceda, A. Pedescoll, R. Gabilondo, A. Prieto-Fernández, and E. Bécares, 2015, Short-term effects of simulated below-ground carbon dioxide leakage on a soil microbial community: International Journal of Greenhouse Gas Control, **36**, 51–59.

Fokker, P. A., E. S. Borello, F. Verga, and D. Viberti, 2018, Harmonic pulse testing for well performance monitoring: Journal of Petroleum Science and Engineering, **162**, 446–459.

Fokker, P. A., and F. Verga, 2011, Application of harmonic pulse testing to water–oil displacement: Journal of Petroleum Science and Engineering, **79**, 125–134.

Gal, F., Z. Pokryszka, N. Labat, K. Michel, S. Lafortune, and A. Marblé, 2019, Soil-gas concentrations and flux monitoring at the lacq-rousse CO_2-geological storage pilot site (french pyrenean foreland): From pre-injection to post-injection: Applied Sciences, **9**, 645.

Gaus, I., 2010, Role and impact of CO_2–rock interactions during CO_2 storage in sedimentary rocks: International journal of greenhouse gas control, **4**, 73–89.

Glumov, N., E. Kolomiyetz, and V. Sergeyev, 1995, Detection of objects on the image using a sliding window mode: Optics & Laser Technology, **27**, 241–249.

Greff, K., R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, 2017, LSTM: A search space odyssey: IEEE Transactions on Neural Networks and Learning Systems, **28**, 2222–2232.

Gu, J., Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al., 2018, Recent advances in convolutional neural networks: Pattern Recognition, **77**, 354–377.

Harris, F. J., 1978, On the use of windows for harmonic analysis with the discrete fourier transform: Proceedings of the IEEE, **66**, 51–83.

Hawthorn, A., S. Aguilar, et al., 2017, New wireless acoustic telemetry system allows real-time downhole data transmission through regular drillpipe: Presented at the SPE Annual Technical Conference and Exhibition, Society of Petroleum Engineers.

Heideman, M., D. Johnson, and C. Burrus, 1984, Gauss and the history of the fast fourier transform: IEEE ASSP Magazine, **1**, 14–21.

Hidaka, A., and T. Kurita, 2017, Consecutive dimensionality reduction by canonical correlation analysis for visualization of convolutional neural networks: , 160–167.

Hochreiter, S., Y. Bengio, P. Frasconi, J. Schmidhuber, et al., 2001, Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

Hosseini, S. A., 2014, Time lapse compressibility monitoring for detection of CO_2 leakage in brine aquifers: Energy Procedia, **63**, 4459–4463.

Hoteit, H., M. Fahs, and M. R. Soltanian, 2019, Assessment of CO_2 injectivity during sequestration in depleted gas reservoirs: Geosciences, **9**, 199.

Ivanova, A., A. Kashubin, N. Juhojuntti, J. Kummerow, J. Henninges, C. Juhlin, S. Lüth, and M. Ivandic, 2012, Monitoring and volumetric estimation of injected CO_2

using 4D seismic, petrophysical data, core measurements and well logging: a case study at ketzin, germany: Geophysical Prospecting, **60**, 957–973.

Jan, B., H. Farman, M. Khan, M. Imran, I. Islam, A. Ahmad, S. Ali, and G. Jeon, 2017, Deep learning in big data analytics: A comparative study: Computers Electrical Engineering.

Jia, Y., E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, 2014, Caffe: Convolutional architecture for fast feature embedding: Proceedings of the 22nd ACM international conference on Multimedia, 675–678.

Johnston, D. H., 2013, Practical applications of time-lapse seismic data: Society of Exploration Geophysicists.

Kandhari, R., V. Chandola, A. Banerjee, V. Kumar, and R. Kandhari, 2009, Anomaly detection: ACM Comput. Surv, **41**, 1–6.

Khan, A., A. Sohail, U. Zahoora, and A. S. Qureshi, 2019, A survey of the recent architectures of deep convolutional neural networks: arXiv preprint arXiv:1901.06032.

Kingma, D. P., and J. Ba, 2014, Adam: A method for stochastic optimization: arXiv preprint arXiv:1412.6980.

LeCun, Y., Y. Bengio, et al., 1995, Convolutional networks for images, speech, and time series: The handbook of brain theory and neural networks, **3361**, 1995.

Macquet, M., D. C. Lawton, J. Donags, and J. Barraza, 2017, Feasibility study of time-lapse-seismic monitoring of CO_2 sequestration: EAGE/SEG Research Workshop 2017, European Association of Geoscientists & Engineers, cp–522.

May, F., and S. Waldmann, 2014, Tasks and challenges of geochemical monitoring: Greenhouse Gases: Science and Technology, **4**, 176–190.

Moore, J., M. Adams, R. Allis, S. Lutz, and S. Rauzi, 2005, Mineralogical and geochemical consequences of the long-term presence of CO_2 in natural reservoirs: an example from the springerville–st. johns field, arizona, and new mexico, usa: Chemical Geology, **217**, 365–385.

Moradi, S., and D. C. Lawton, 2013, Theoretical detectability of CO_2 at a CCS project in alberta, *in* SEG Technical Program Expanded Abstracts 2013: Society of Exploration Geophysicists, 3475–3479.

Mozer, M. C., 1995, A focused backpropagation algorithm for temporal: Backpropagation: Theory, architectures, and applications, **137**.

Nguyen, M., T. He, L. An, D. C. Alexander, J. Feng, B. T. Yeo, A. D. N. Initiative, et al., 2019, Predicting alzheimer's disease progression using deep recurrent neural networks: BioRxiv, 755058.

Nwankpa, C., W. Ijomah, A. Gachagan, and S. Marshall, 2018, Activation functions: Comparison of trends in practice and research for deep learning: arXiv preprint arXiv:1811.03378.

Oelkers, E. H., S. R. Gislason, and J. Matter, 2008, Mineral carbonation of CO_2: Elements, **4**, 333–337.

Olivier, G., M. Chmiel, F. Brenguier, P. Roux, A. Mordret, P. Dales, T. Lecocq, and I. Djamaludin, 2018, Processing passive seismic data recorded on a dense array for CCS site characterization, *in* SEG Technical Program Expanded Abstracts 2018: Society of Exploration Geophysicists, 3002–3006.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., 2011, Scikit-learn: Machine learning in Python: the Journal of machine Learning research, **12**, 2825–2830.

Reeves, M. E., P. L. Camwell, J. McRory, et al., 2011, High speed acoustic telemetry network enables real-time along string measurements, greatly reducing drilling risk: Presented at the Offshore Europe, Society of Petroleum Engineers.

Roach, L. A., D. J. White, and B. Roberts, 2015, Assessment of 4D seismic repeatability and CO_2 detection limits using a sparse permanent land array at the aquistore CO_2 storage site: Geophysics, **80**, WA1–WA13.

Rochelle, C. A., I. Czernichowski-Lauriol, and A. Milodowski, 2004, The impact of chemical reactions on CO_2 storage in geological formations: a brief review: Geological Society, London, Special Publications, **233**, 87–106.

Rutqvist, J., 2012, The geomechanics of CO_2 storage in deep sedimentary formations: Geotechnical and Geological Engineering, **30**, 525–551.

Rynkiewicz, J., 2012, General bound of overfitting for MLP regression models: Neurocomputing, **90**, 106–110.

Sathyanarayana, S., 2014, A gentle introduction to backpropagation: Numeric Insight, **7**, 1–15.

Selma, L., O. Seigo, S. Dohle, and M. Siegrist, 2014, Public perception of carbon capture and storage (CCS): A review: Renewable and Sustainable Energy Reviews, **38**, 848–863.

Shao, H., D. A. Ussiri, C. G. Patterson, R. A. Locke II, H. Wang, A. H. Taylor, and H. F. Cohen, 2019a, Soil gas monitoring at the illinois basin–decatur project carbon sequestration site: International Journal of Greenhouse Gas Control, **86**, 112–124.

———, 2019b, Soil gas monitoring at the illinois basin–decatur project carbon sequestration site: International Journal of Greenhouse Gas Control, **86**, 112–124.

Simonyan, K., and A. Zisserman, 2014, Very deep convolutional networks for large-scale image recognition: arXiv preprint arXiv:1409.1556.

Smith, K., M. Steven, D. Jones, J. West, P. Coombs, K. Green, T. Barlow, N. Breward, S. Gwosdz, M. Krüger, S. Beaubien, A. Annunziatellis, S. Graziani, and S. Lombardi, 2013, Environmental impacts of CO_2 leakage: recent results from the ASGARD facility, UK: Energy Procedia, **37**, 791–799.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, 2014, Dropout: a simple way to prevent neural networks from overfitting: The journal of machine learning research, **15**, 1929–1958.

Stork, A. L., C. Allmark, A. Curtis, J.-M. Kendall, and D. J. White, 2018, Assessing the potential to use repeated ambient noise seismic tomography to detect CO_2 leaks: Application to the aquistore storage site: International Journal of Greenhouse Gas Control, **71**, 20–35.

Sun, A. Y., A. Kianinejad, J. Lu, and S. Hovorka, 2014a, A frequency-domain diagnosis tool for early leakage detection at geologic carbon sequestration sites: Energy Procedia, **63**, 4051–4061.

———, 2014b, A frequency-domain diagnosis tool for early leakage detection at geologic carbon sequestration sites: Energy Procedia, **63**, 4051–4061.

Sun, A. Y., J. Lu, B. M. Freifeld, S. D. Hovorka, and A. Islam, 2016, Using pulse testing for leakage detection in carbon storage reservoirs: A field demonstration: International Journal of Greenhouse Gas Control, **46**, 215–227.

Vafaeipour, M., O. Rahbari, M. A. Rosen, F. Fazelpour, and P. Ansarirad, 2014, Application of sliding window technique for prediction of wind velocity time series: International Journal of Energy and Environmental Engineering, **5**, 105.

Verdon, J. P., J.-M. Kendall, and S. C. Maxwell, 2010a, A comparison of passive seismic monitoring of fracture stimulation from water and CO_2 injection: Geophysics, **75**, MA1–MA7.

———, 2010b, A comparison of passive seismic monitoring of fracture stimulation from water and CO_2 injection: Geophysics, **75**, MA1–MA7.

Volkova, S., T. Wilson, and D. Yarowsky, 2013, Exploring demographic language variations to improve multilingual sentiment analysis in social media: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 1815–1827.

Wu, X., V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip, et al., 2008, Top 10 algorithms in data mining: Knowledge and information systems, **14**, 1–37.

Xingjian, S., Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, 2015a, Convolutional LSTM network: A machine learning approach for precipitation nowcasting: Advances in neural information processing systems, 802–810.

———, 2015b, Convolutional LSTM network: A machine learning approach for precipitation nowcasting: Advances in neural information processing systems, 802–810.

Yilmaz, I., and O. Kaynar, 2011, Multiple regression, ANN (RBF, MLP) and ANFIS models for prediction of swell potential of clayey soils: Expert systems with applications, **38**, 5958–5966.

Zeiler, M. D., and R. Fergus, 2014, Visualizing and understanding convolutional networks: European conference on computer vision, Springer, 818–833.

Zhong, Z., A. Sun, Y. Qian, and Q. Ouyang, 2019a, A deep learning approach to anomaly detection in geological carbon sequestration sites using pressure measurements: Journal of Hydrology, **573**.

Zhong, Z., A. Y. Sun, Q. Yang, and Q. Ouyang, 2019b, A deep learning approach to anomaly detection in geological carbon sequestration sites using pressure measurements: Journal of hydrology, **573**, 885–894.

Zhou, Z., Y. Lin, Z. Zhang, Y. Wu, Z. Wang, R. Dilmore, and G. Guthrie, 2019, A data-driven CO_2 leakage detection using seismic data and spatial–temporal densely connected convolutional neural networks: International Journal of Greenhouse Gas Control, **90**, 102790.

Zimek, A., and P. Filzmoser, 2018, There and back again: Outlier detection between statistical reasoning and data mining algorithms: Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, **8**, e1280.