Time-lapse monitoring using a stacked bidirectional long short-term memory neural network

Shang Huang and Daniel O. Trad

ABSTRACT

Time-lapse seismic monitoring quality is affected by near-surface noise, weak reservoir change amplitude and poor subsurface illumination. Except for some geophysical approaches, deep learning can solve the challenges above with high efficiency and accuracy. This project uses a stacked bidirectional long short-term memory neural network (SD-Bi-LSTM) to predict near-surface noise from baseline seismic data. Furthermore, the surface multiple is added in forward modeling to generate baseline and monitor data with expanded subsurface illumination. Results show that stacked bidirectional long short-term memory can predict and mitigate noise in monitor data. The final difference between baseline and monitor models has suppressed significant noise after combining SD-Bi-LSTM and surface multiples. Images have improved accuracy and quality.

INTRODUCTION

Time-lapse seismic monitoring, which acquires seismic data at different times over the same site to obtain fluid-flow variations, has contributed to detecting subsurface physical properties and reservoir behavior in recent years (Wang, 1997; Koster et al., 2000; Pennington, 2000; Lumley, 2001; Arts et al., 2003; Isaac and Lawton, 2006, 2014; Chadwick et al., 2010; Wang and Morozov, 2020; Henley and Lawton, 2021). Particularly, time-lapse in investigating carbon storage reservoirs becomes important, because monitoring CO_2 migration status is essential for carbon capture and storage, which will help to locate oil and gas displacement effects during long-term change. It can improve to estimate secondary recovery of reservoir, or fluid injection.

Currently, there are some challenges for time-lapse seismic monitoring. Seismic imaging of weak time-lapse changes remains a major challenge among others because weak change's amplitude might be covered by other noise generated by near surface, which is hard to distinguish. Excepting that weak reservoir change amplitude is affected by artifacts, time-lapse monitoring analysis may construct false anomalies due to poor subsurface illumination and the inaccurate input image. Additionally, 4d seismic may only capture some reservoir change geometry. Some geophysical approaches (Rickett and Lumley, 2001; Ayeni and Biondi, 2010; Zhang et al., 2013; Bergmann et al., 2014; Wapenaar and Van Ijsseldijk, 2021; Fu and Innanen, 2022) try to solve the challenges above. Except for that, deep neural networks have become prevalent methods to deal with time-lapse monitoring of carbon capture and storage with high efficiency and accuracy (Yuan et al., 2020; Zhong et al., 2020; Li et al., 2021; Hussein et al., 2021; Alali et al., 2022; Li and Alkhalifah, 2022). Long short-term memory (LSTM) and bidirectional long short-term memory (Bi-LSTM) (Hochreiter and Schmidhuber, 1997; Graves and Schmidhuber, 2005), as two typical recurrent neural networks have been applied in many geophysical problems to learn non-linear relationships. For example, seismic data reconstruction (Yoon et al., 2020), missing well log estimation (Pham and Wu, 2019), elastic properties and litho-fluid facies estimation

(Aleardi, 2022), seismic impedance inversion and parameter estimation (Calderón-Macías et al., 2000; Moya and Irikura, 2010; Alfarraj and AlRegib, 2019; Das et al., 2019; Guo et al., 2019; Roy et al., 2020). For Bi-LSTM, it can learn from both long-term forward and backward temporal dependencies from historical data, and it works with long and dense temporal traces. Thus, this method is suitable to be used in time-lapse seismic data.

In this project, we propose a data-driven method to predict baseline data with noise from monitor shot traces, using stacked bidirectional long short-term memory (SD-Bi-LSTM) and surface multiple reflections. Stacked bidirectional long short-term memory is used to predict noise variation of monitor seismic data from baseline data. Stacked layers' mechanisms can enhance the power of neural networks. The seismic data is obtained from elastic reverse time migration using GPU. By this approach, reservoir change will be predicted properly with artifact suppression. Additionally, we take advantage of multiple reflections generated from free surface boundary conditions. It can help with broadening subsurface illumination and indicate accurate reservoir variation location.

THEORY

Time-lapse seismic

In time-lapse seismic, base data is set as the reference. After injecting CO2 or other fluids into a reservoir, monitor data is obtained to indicate the seismic attribute variations. Based on Alali et al. (2022), the residual between monitor data $\mathbf{d}_{obs_m}$ and base data $\mathbf{d}_{obs_b}$ is

$$\delta \mathbf{d}(t) = \mathbf{d}_{obs_m}(t) - \mathbf{d}_{obs_b}(t) = \mathbf{n}(t) + \delta \mathbf{r}(t) \tag{1}$$

where t means time, $\mathbf{n}(t)$ denotes noise generated from near-surface change and non-repeatable signal, which should be eliminated. The data difference term, $\delta\mathbf{r}(t)$, is our target is detect subsurface reservoir variation. Manually removing noise is hard to achieve, whereas neural networks can learn the patterns from the noisy part in monitor data and mitigate the noise. Then, reservoir variation can be obtained from the difference between monitor data and predicted base data.

Recurrent neural network (RNN) and long short-term memory (LSTM)

A recurrent neural network (RNN) is an artificial neural network that uses sequential or time series data. Recurrent neural network (RNN) (Jordan, 1986; Rumelhart et al., 1985) is derived from the a feed-forward neural network where the connections between nodes do not form a cycle and deliver information in one direction (Figure 1). On the opposite, RNN has an internal self-looped deep-learning architecture. It allows previous output to affect subsequent input and output. In other words, the current input will learn from and depend on the past sequence output. After obtaining the current output, it will be sent back into the recurrent network. The benefit of RNN is that it can process variable-length sequences of input, for example, time sequences. While training an RNN model, vanishing or exploding gradient issues might occur. If the gradient is too small or large, it tends to grow or vanish when it is passed back through many time steps.

Long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) was designed with special memory cells to store temporal information. It can remember values over arbitrary time intervals with gate structures shown in Figure 2. Also, it can avoid the vanishing gradient problem usually occurring in RNN. The gradient in LSTM contains the gate's vector of activations, allowing the network to control the gradient values better and avoid getting too small or large. This structure allows LSTM to remember long-range features better than conventional recurrent neural networks. Similar to time series, the seismic data traces also have long-term and dense sampling rates. Because LSTM can capture very long-term dependencies, it is also suitable for working with seismic traces.

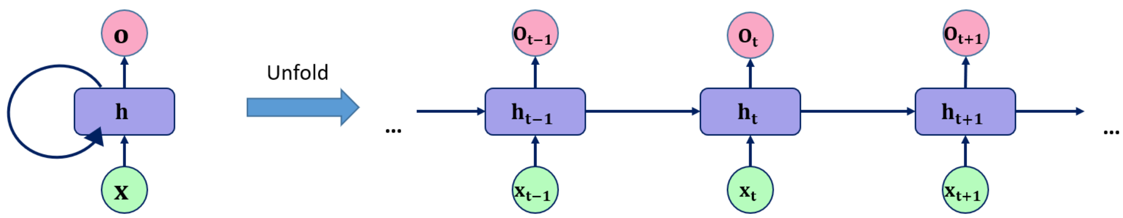


FIG. 1: Workflow for recurrent neural network (RNN).

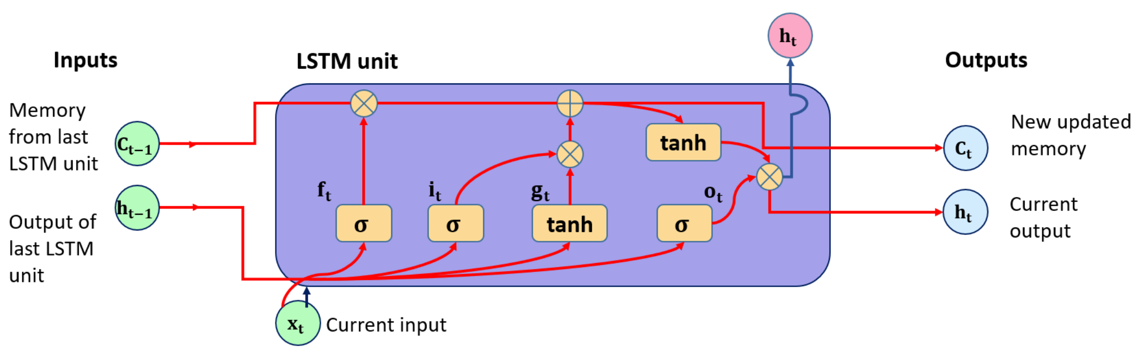


FIG. 2: Workflow for long short-term memory (LSTM) algorithm.

We can start with a simple LSTM framework, then dive into the BLSTM that was used in this project. Within each LSTM cell (shown in Figure 2), there are four gates in total: $\mathbf{f}_t$, $\mathbf{i}_t$, $\mathbf{g}_t$ and $\mathbf{o}_t$ are respectively the *forget gate*, *input gate*, *candidate gate* and *output gate* cell activation vectors. They have the same size as the hidden vector $\mathbf{h}_t$. Next, each gate vector will be illustrated in detail. The forget gate f_t is determined by

$$\mathbf{f}_t = \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \tag{2}$$

where $\mathbf{h}_{t-1}$ is the hidden layer vector from previous time and $\mathbf{x}_t$ means the current input vector. σ, $\mathbf{W}_f$ and $\mathbf{b}_f$ represent the logistic sigmoid activation function, weight matrices and bias for the forget gate.

Then, the input gate follows a similar behavior as forget gate but with different weight and bias in the input gate i_t:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \tag{3}$$

Another gate named candidate gate g_t can be determined by:

$$\mathbf{g}_t = tanh(\mathbf{W}_g[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_g) \tag{4}$$

where $tanh()$ denote a tanh activation layer.

The last gate in the LSTM unit is output gate o_t, and it can be obtained by:

$$\mathbf{o}_t = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \tag{5}$$

After having all the gates, the next step is to calculate and determine a new updated memory C_t and current output h_t. The former needs memory from last LSTM unit C_{t-1} combined with forget gate f_t and the product of input gate i_t and candidate gate g_t. Then, the current cell state C_t can be calculated by

$$\mathbf{C}_t = \mathbf{i}_t\mathbf{g}_t + \mathbf{f}_t\mathbf{C}_{t-1} \tag{6}$$

As for the output of the LSTM cell h_t, it is made of the product of output gate o_t and current cell state C_t after applying tanh activation layer:

$$\mathbf{h}_t = \mathbf{o}_t tanh(\mathbf{C}_t) \tag{7}$$

Bidirectional long short-term memory (Bi-LSTM)

Unlike conventional RNNs make use of previous contexts, bidirectional RNNs can deal with sequential data in both directions: forward and backward with two separate hidden layers (Schuster and Paliwal, 1997; Graves et al., 2013b). Based on BRNNs and LSTM, bidirectional long short-term memory (Graves and Schmidhuber, 2005) was developed to capture long sequences in reverse and forward two directions. One hidden layer processes the input sequence in the forward direction. This flow does a forward pass for forward and backward states, and the output layer. The other hidden layer handles the input in the reverse direction. It includes doing backward passes for output neurons, forward and backward states, and updating error functions and weights. Thus, the output of the current time step is obtained from both layers' hidden vectors.

The forward function of Bi-LSTM with inputs of M units with N hidden units is shown below:

$$\mathbf{h}_t^n = \sum_{m=1}^{M} x_t^n w_{mn} + \sum_{n'=1,t>0}^{N} \alpha_{t-1}^{n'} w_{n'n} \tag{8}$$

$$\alpha_t^n = \Theta_n(h_t^n) \tag{9}$$

where h_t^n is the network input, and x_t denotes the sequence input. w_{mn} represents the weight of the input m to hidden unit n, and $w_{n'n}$ means the weight of hidden unit n towards hidden unit n'. The activation function of hidden unit n at time step t is given by α_t^n. Θ_n means the activation function of the hidden unit n.

As for the backward calculation, it is

$$\frac{\delta O}{\delta w_n k} = \sum_{t=1}^{T} \frac{\delta O}{\delta h_t^n} \alpha_t^n \tag{10}$$

$$\frac{\delta O}{\delta \alpha_t^n} = \Theta_n'(h_t^n)(\sum_{k=1}^{K} \frac{\delta O}{\delta h_t^n} w_{nk} + \sum_{n'=1,t>0}^{H} \frac{\delta O}{\delta h_{t+1}^{n'}} w_{nn'}) \tag{11}$$

where O denotes an objective function with unit K output.

For the bidirectional LSTM, forward LSTM and backward LSTM need to consider as two separate layers (Du et al., 2020). The final output can be obtained by

$$h_t = \alpha h_t^f + \beta h_t^b \tag{12}$$

$$y_t = \sigma(h_t) \tag{13}$$

where h_t^f is the forward LSTM layer output which takes time sequences from x_1 to x_T, h_t^b denotes the backward LSTM layer output which takes the reverse time sequences from x_T to x_1. α and β represent the importance of forward LSTM and backward LSTM, and satisfy $\alpha + \beta = 1$. h_t is the sum of two LSTM outputs, and y_t is the impedance prediction. Note that a linear activation function is chosen for the dense output layer because seismic attributes might have negative values.

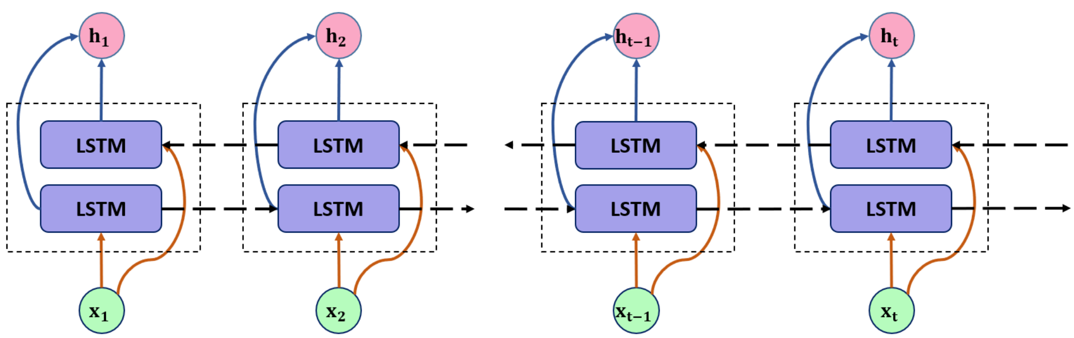


FIG. 3: Workflow for bidirectional long short-term memory (Bi-LSTM) algorithm.

Stacked bidirectional long short-term memory (SD-Bi-LSTM)

Based on the research from Graves et al. (2013a), and Cui et al. (2018), deep bidirectional long short-term memory can be generated by stacking several Bi-LSTM hidden layers on top of each one. The output of one Bi-LSTM hidden layer will be fed into the subsequent Bi-LSTM hidden layer as the input. Deep bidirectional RNNs can be implemented by using the forward $\overrightarrow{h}^n$ and backward sequences $\overleftarrow{h}^n$.Figure 4 shows the stacked bidirectional long short-term memory mechanism. The input of every hidden layer should consist of both the forward and backward layers at the level below. Stacked Bi-LSTM can detect and build up an effectively high level of sequential data representations.

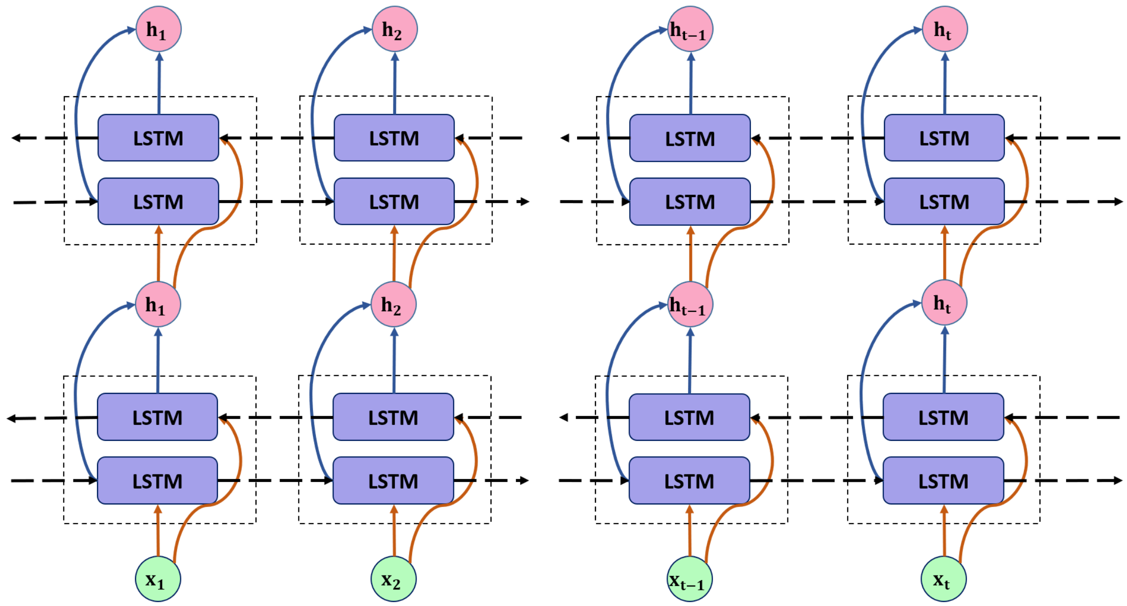


FIG. 4: Stacked bidirectional long short-term memory (Bi-LSTM).

WORKFLOW

The workflow in this project, shown in Figure 5, includes three steps: first is training stacked bidirectional long short-term memory neural networks by using different traces with direct arrivals. We did not remove direct arrivals because it can help mitigate near-surface noise. Traces are separated into two parts for two training goals: simulate near surface noise, and mimic reservoir changes. After that, using stored neural networks to predict baseline and monitor data. The final step is to calculate the migrated image difference between monitor data and predicted baseline data to determine time-lapse reservoir changes.

Measurements

Mean squared error (MSE)

The mean squared error (MSE) loss is applied to evaluate the model performance and penalize the large prediction errors:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{d}^i_{calc} - \mathbf{d}^i_{obs})^2, \tag{14}$$

where n is the total number of samples, $\mathbf{d}_{calc}$ is predicted monitor or baseline data, and $\mathbf{d}_{obs}$ denotes the corresponding observed monitor or baseline data.

Peak signal-to-noise ratio (PSNR)

A peak signal-to-noise ratio (PSNR) is used to evaluate the model performance:

$$\text{PSNR} = 20 * \log_{10}(\frac{\text{MAX}_I}{\sqrt{\text{MSE}}}), \tag{15}$$

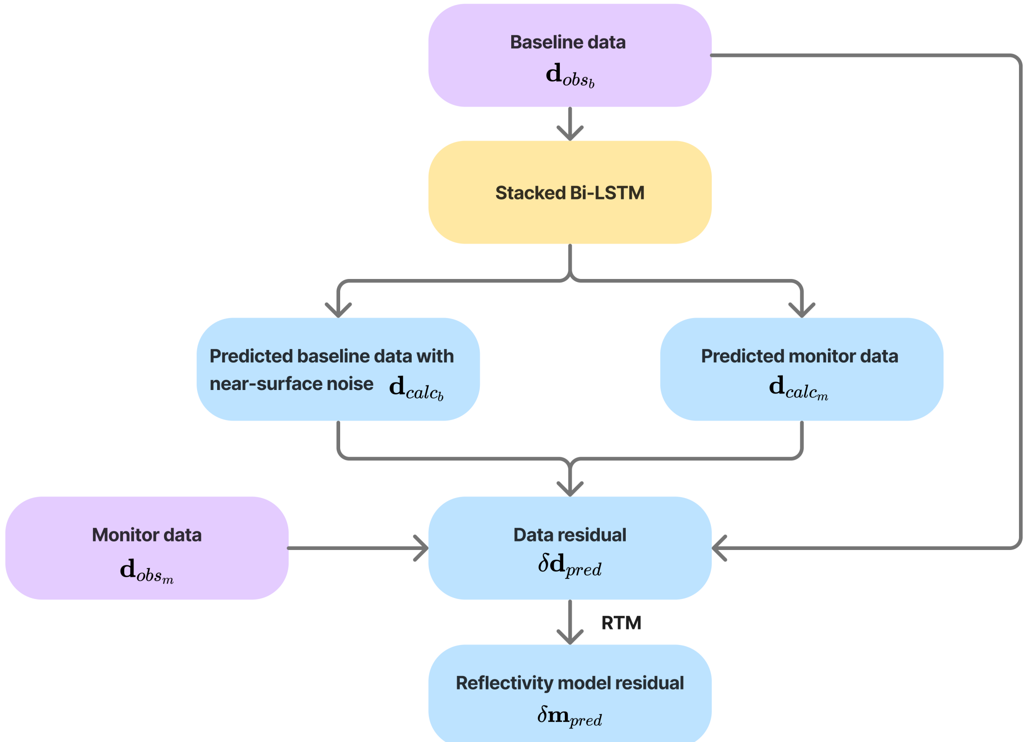
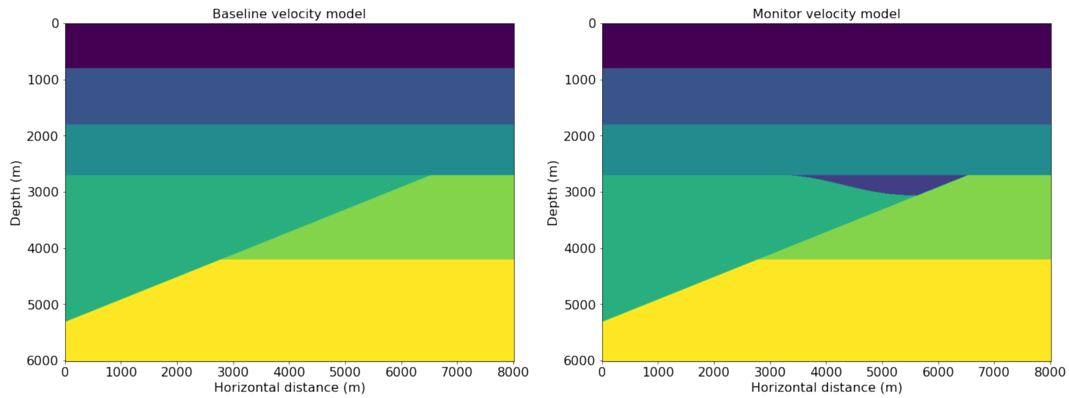


FIG. 5: Workflow for predicting time-lapse reflectivity change.

where MAX_I denotes the maximum possible pixel value of the image, and MSE is the mean squared error based on the equation 14.

TRAIN, TEST AND VALIDATION

One assumption in this project is that monitor and baseline systems are the same for basic training purposes. In future work, we will test different systems, or add near-surface velocity changes. A horizontal-layered with a dipping event geology model is used for training and testing. A slight reservoir anomaly is located at 2800 meters depth to generate weak amplitude data difference. Monitor and baseline velocity models are shown in Figure 6 whose size is 601×801 grid points. There are 16 shots simulated on the surface with 500 meters spatial interval. The number of receivers is the same as horizontal points, which is 801. The starting point is located at 50 meters horizontal distance. The temporal and spatial sampling rates are 1 ms and 10 meters, respectively. Bidirectional long short-term memory is utilized to predict monitor data traces from base survey seismic data. Thus, the base data is considered the input of the neural network model. However, unlike training the whole base data at one time, we extend the idea of Alali et al. (2022), the input traces are divided into two parts: traces collected near the surface, and the other part includes reflections above the CO2 injection area. Data chosen from the near surface is used for noise simulation. A neural network trained by the second part of data can learn other artifacts or noise generated from monitor data.



(a) (b)

FIG. 6: Velocity model for (a) baseline and (b) monitor systems.

At this stage, to speed up the training process, half the number of shots are implemented in the training whose training and testing sets rate is 0.8: 0.2; the rest is for validation. In the training process, we developed two neural networks. One is only applied in the first part data to learn the difference between base and monitor data in the shallow depth or near the surface. This process helps to predict noise and suppress it in the migration image. Then, pseudo monitor data traces for deeper structure, but above the reservoir variation can be predicted by the other trained neural network with the second data part. The residual between observed monitor data and predicted traces indicates CO2 injection information with less noise, because the neural network does not learn how to differentiate reservoir changes from the base survey.

NUMERICAL EXAMPLE

In this section, we implement the same geology model, which is used in the training and testing process, with different source-receiver locations. The start location for the shot is 300 meters with 750 meters spatial interval. Seven shots are fed into two trained neural networks separately with different data parts. After prediction, an example of comparison between observed and calculated data is shown in Figure 7. It is extracted from the third shot at a time window between 2.5 and 4.0 seconds. Figure 7a and b represent baseline data $\mathbf{d}_{obs_b}$ and monitor data $\mathbf{d}_{obs_m}$, and Figure 7c and d denote predicted baseline $\mathbf{d}_{calc_b}$ and monitor $\mathbf{d}_{calc_m}$ data. Since neural networks are trained based on data above the reservoir change, the original baseline and predicted baseline and monitor data share a similar data pattern that does not contain anomalies. However, the relative amplitude in predicted baseline data $\mathbf{d}_{calc_b}$ is stronger between 3.6 and 4.0 seconds, which might be subtracted from monitor data to suppress noise in migration.

The data residual between observation and prediction is shown in Figure 8.The left figure gives the observed difference ($\mathbf{d}_{obs_m} - \mathbf{d}_{obs_b}$), and the right one means the calculated difference between observed monitor and predicted baseline data ($\mathbf{d}_{obs_m} - \mathbf{d}_{calc_b}$). Except for reservoir data changes, the calculated difference gives additional information for direct arrivals and other noise at both offset sides. For a detailed comparison, Figure 9 delivers a trace comparison at 6000 meters horizontal distance between different data. Predicted

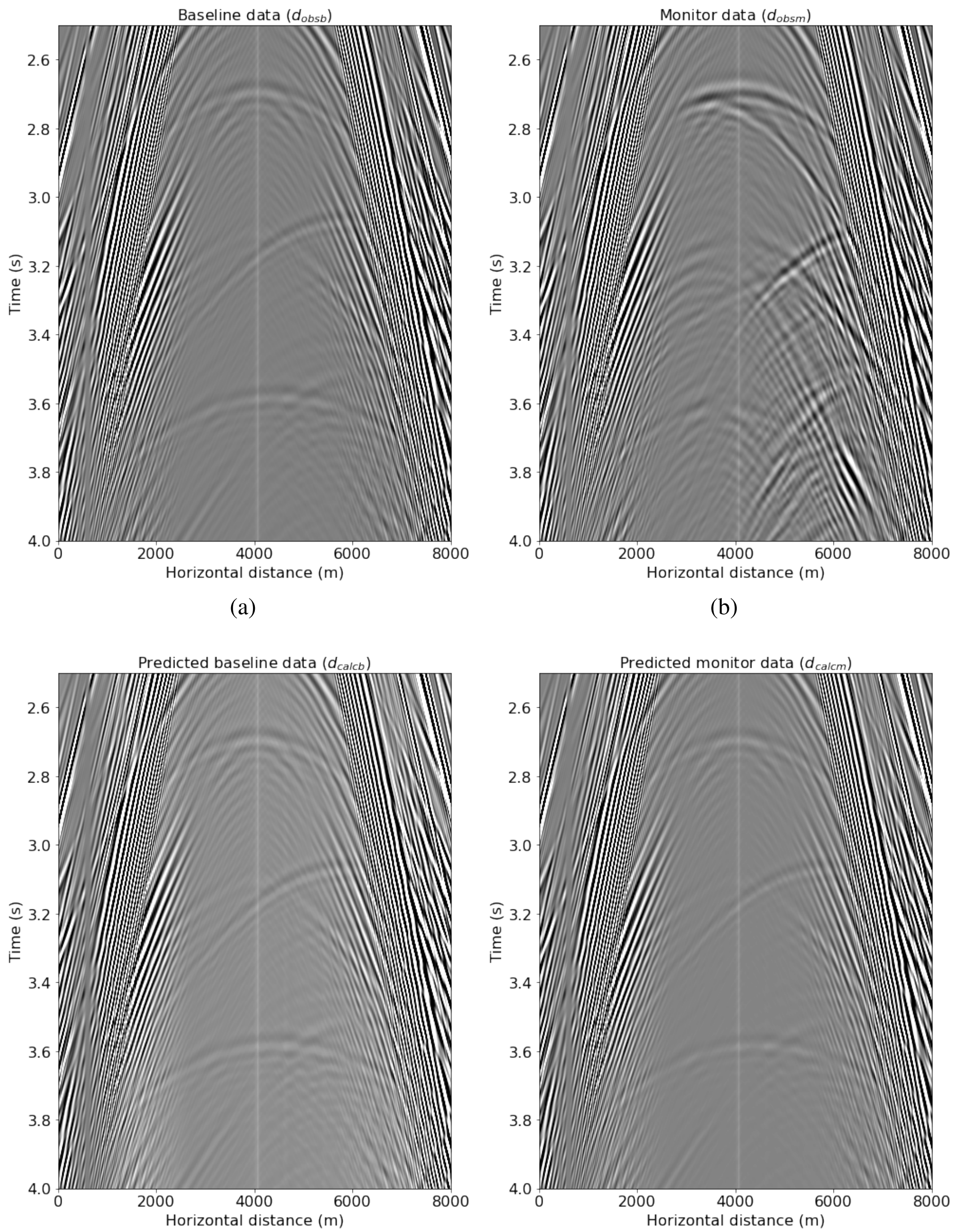
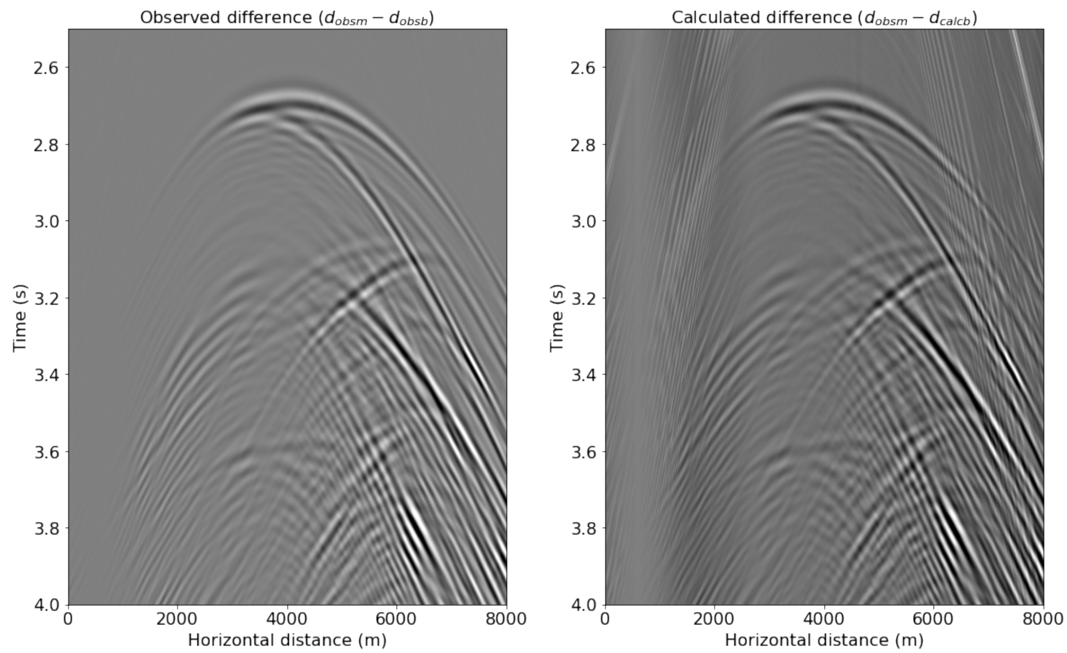


FIG. 7: Data obtained from (a) observed baseline, (b) observed monitor, (c) predicted baseline and (d) predicted monitor system.

baseline data (solid red line) keeps a basic pattern as observed baseline (green starred line), but also recovers amplitudes for some events that are close to observed monitor data (black dashed line). As for the predicted monitor data (blue dashed line), its amplitude is more stable than the predicted baseline data. The reason is that predicted baseline data, which is predicted from the shallow depth data with noise, whereas the predicted monitor data is derived from relatively deep data above reservoir change, which has a regular pattern to learn for the neural network.



(a) (b)

FIG. 8: Data difference between (a) observed monitor data and baseline data, and (b) observed monitor data and predicted baseline data.

Migration results from observed and predicted data are shown in Figure 10 and 11. Migration images of predicted baseline (Figure 10c) and monitor data (Figure 10d) have increased amplitude for events above the reservoir at 2800 meters depth. To obtain migration difference, we implement the double difference concept that originated from tomography and inversion (Watanabe et al., 2004; Asnaashari et al., 2015; Zhou et al., 2010). The difference between the two sets of data is:

$$\delta\mathbf{d} = (\mathbf{d}_{obs_m} - \mathbf{d}_{obs_b}) - (\mathbf{d}_{calc_m} - \mathbf{d}_{calc_b}) \tag{16}$$

where $\mathbf{d}_{obs_m}$ and $\mathbf{d}_{obs_b}$ denote observed monitor and baseline data separately, and $\mathbf{d}_{calc_m}$ and $\mathbf{d}_{calc_b}$ give predicted data. Thus, the time-lapse model changes $\delta\mathbf{m}_{time-lapse}$ can be derived:

$$\delta\mathbf{m}_{time-lapse} = \delta\mathbf{m}_{obs} - \delta\mathbf{m}_{calc} \tag{17}$$

where $\delta\mathbf{m}_{obs}$ and $\delta\mathbf{m}_{calc}$ are calculated from the first and second parentheses in equation 16, respectively.

Then, calculated double-difference model residual $\delta\mathbf{m}_{time-lapse}$ is obtained and shown in Figure 11a. Compared with $\delta\mathbf{m}_{obs}$ (Figure 11c) and $\delta\mathbf{m}_{calc}$ (Figure 11b), $\delta\mathbf{m}_{time-lapse}$ has suppressed artifacts above the reservoir change, which is shown in red arrow. Additionally, amplitude and illumination for the anomaly are more amplified in Figure 11a than the other two results.

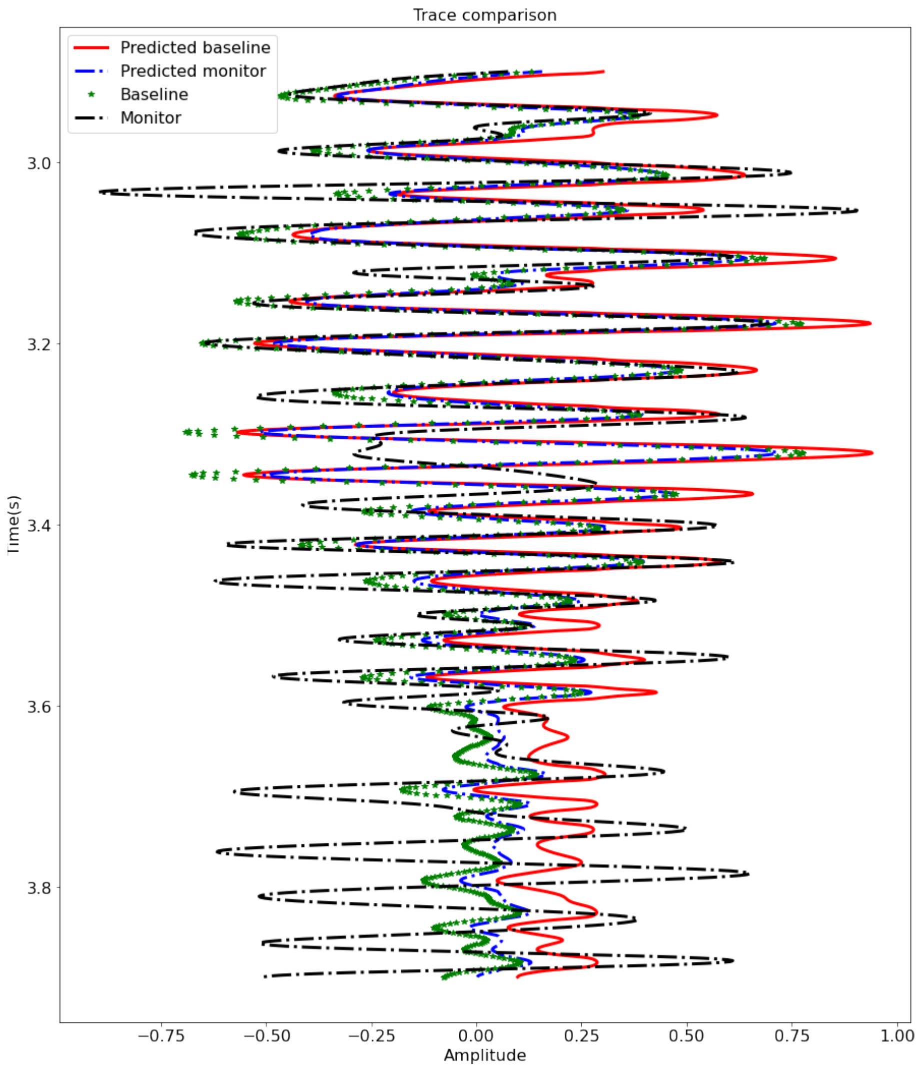


FIG. 9: Trace comparison at 6000 meters horizontal distance.

The peak signal-to-noise ratio of double difference model residual has the highest value, 21.12 dB compared with other two model residuals indicated in Table 1. It also proves that our proposed method gives improved accuracy and image quality.

Figure 12 and 13 indicate train and validation loss of the first and second neural networks, respectively. Note that validation loss in both models is smaller than training loss along all the iterations. Even though this situation is common in long short-term memory neural network training, one possible reason is that the train and test data size is small-scale, which might lead to overfitting or underfitting. Future work will consider enlarging the train and test dataset.

Table 1: PSNR (dB) comparison different model residuals			
Prediction	Original $\delta\mathbf{m}_{obs}$	Predicted $\delta\mathbf{m}_{calc}$	**Double difference** $\delta\mathbf{m}_{time-lapse}$
Example	15.19	15.26	**21.12**

Table 1: PSNR (dB) comparison different model residuals.

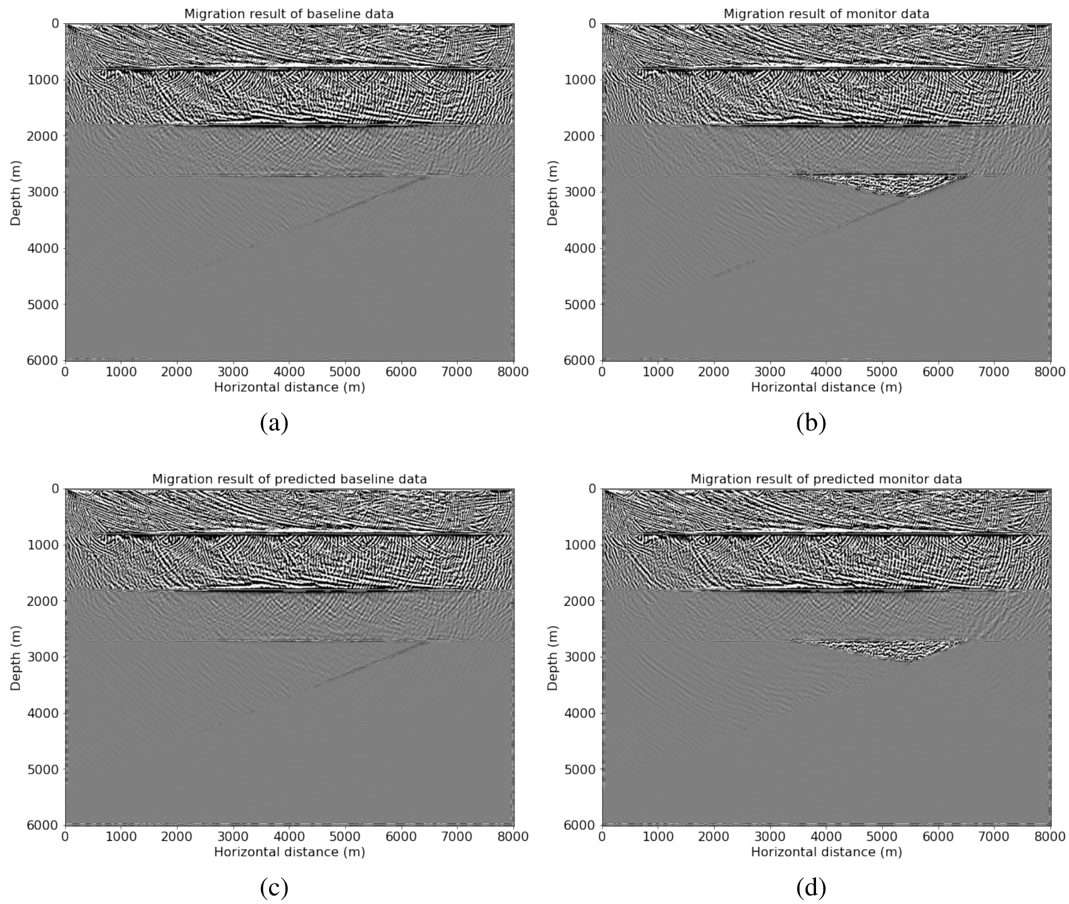


FIG. 10: Migrated result from (a) observed baseline, (b) observed monitor, (c) predicted baseline and (d) predicted monitor system.

CONCLUSIONS AND FUTURE WORK

In this project, we proposed to use two stacked bidirectional long short-term memory neural networks to learn near-surface noise and shallow depth signal information of monitor data. Predicted baseline and monitor data can recover and mimic noise patterns that will be subtracted from observed monitor data using the double-difference method, to mitigate noise. Results show that migration image by this method has improved artifact suppression with significant amplitude for reservoir change. In future work, we will try different complex geology models to train and test this neural network architecture and workflow to make it generalized to other situations.

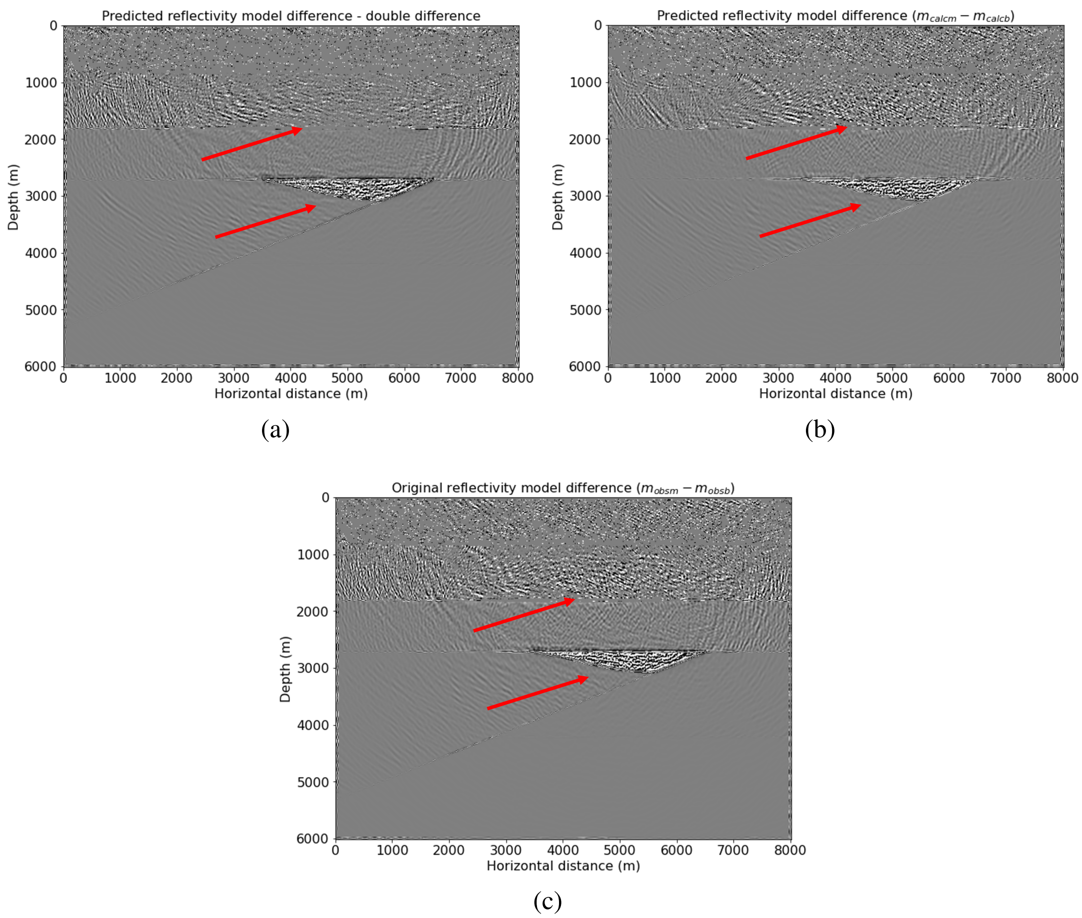


FIG. 11: Migrated result comparison between (a) model difference from double-difference method, (b) predicted model difference and (c) original model difference.

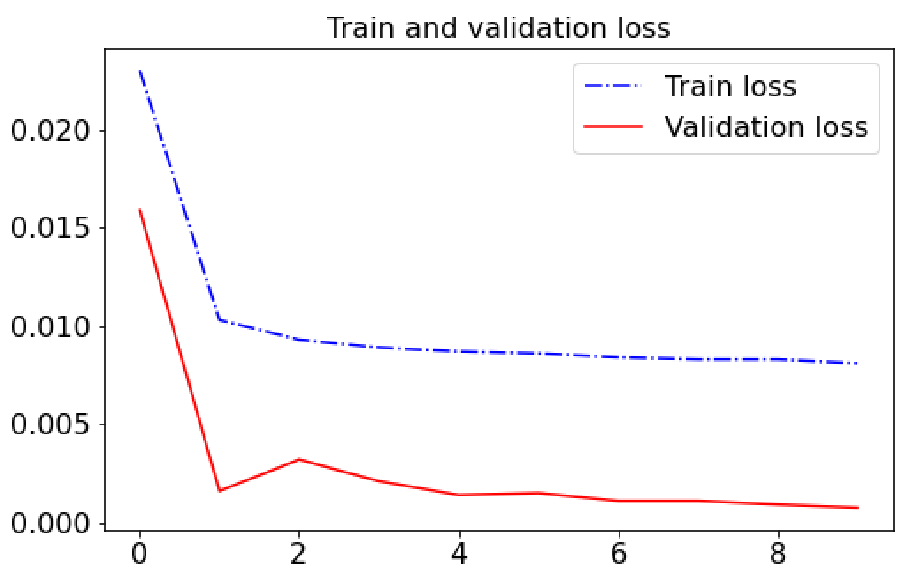


FIG. 12: Train and validation loss.

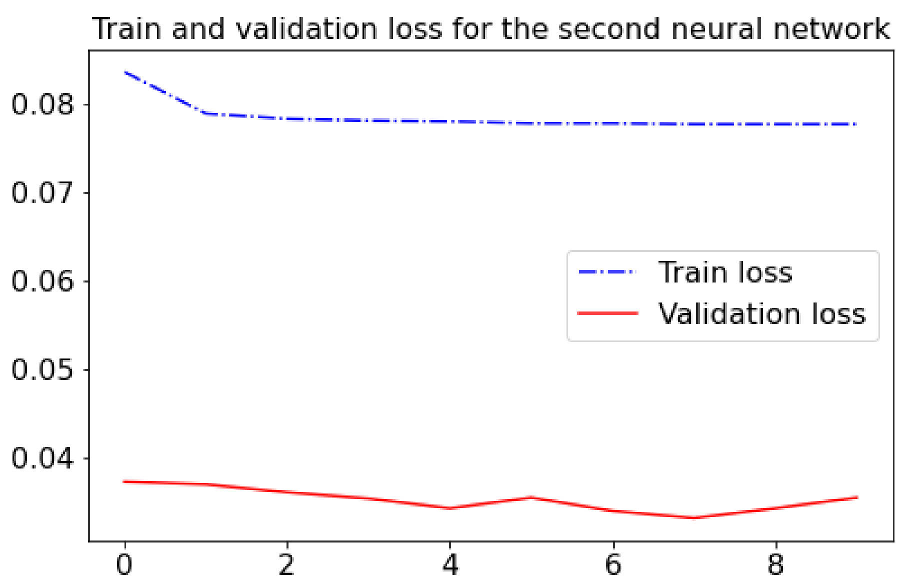


FIG. 13: Train and validation loss for the second neural network.

ACKNOWLEDGEMENTS

We thank the sponsors of CREWES for continued support. This work was funded by CREWES industrial sponsors and NSERC (Natural Science and Engineering Research Council of Canada) through the grant CRDPJ 543578-19. The first author was partially supported by a scholarship from the China Scholarship Council (CSC).

REFERENCES

Alali, A., Kazei, V., Sun, B., and Alkhalifah, T., 2022, Time-lapse data matching using a recurrent neural network approach: Geophysics, **87**, No. 5, 1–83.

Aleardi, M., 2022, Elastic properties and litho-fluid facies estimation from pre-stack seismic data through bidirectional long short-term memory networks: Geophysical Prospecting, **70**, No. 3, 558–577.

Alfarraj, M., and AlRegib, G., 2019, Semisupervised sequence modeling for elastic impedance inversion: Interpretation, **7**, No. 3, SE237–SE249.

Arts, R., Eiken, O., Chadwick, A., Zweigel, P., Van der Meer, L., and Zinszner, B., 2003, Monitoring of co2 injected at sleipner using time lapse seismic data, *in* Greenhouse Gas Control Technologies-6th International Conference, Elsevier, 347–352.

Asnaashari, A., Brossier, R., Garambois, S., Audebert, F., Thore, P., and Virieux, J., 2015, Time-lapse seismic imaging using regularized full-waveform inversion with a prior model: which strategy?: Geophysical prospecting, **63**, No. 1, 78–98.

Ayeni, G., and Biondi, B., 2010, Target-oriented joint least-squares migration/inversion of time-lapse seismic data sets: Geophysics, **75**, No. 3, R61–R73.

Bergmann, P., Kashubin, A., Ivandic, M., Lüth, S., and Juhlin, C., 2014, Time-lapse difference static correction using prestack crosscorrelations: 4d seismic image enhancement case from ketzin: Geophysics, **79**, No. 6, B243–B252.

Calderón-Macías, C., Sen, M. K., and Stoffa, P. L., 2000, Artificial neural networks for parameter estimation in geophysics [link]: Geophysical prospecting, **48**, No. 1, 21–47.

Chadwick, A., Williams, G., Delepine, N., Clochard, V., Labat, K., Sturton, S., Buddensiek, M.-L., Dillen, M., Nickel, M., Lima, A. L. et al., 2010, Quantitative analysis of time-lapse seismic monitoring data at the sleipner co 2 storage operation: The Leading Edge, **29**, No. 2, 170–177.

Cui, Z., Ke, R., Pu, Z., and Wang, Y., 2018, Deep bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction: arXiv preprint arXiv:1801.02143.

Das, V., Pollack, A., Wollner, U., and Mukerji, T., 2019, Convolutional neural network for seismic impedance inversion: Geophysics, **84**, No. 6, R869–R880.

Du, G., Wang, Z., Gao, B., Mumtaz, S., Abualnaja, K. M., and Du, C., 2020, A convolution bidirectional long short-term memory neural network for driver emotion recognition: IEEE Transactions on Intelligent Transportation Systems.

Fu, X., and Innanen, K. A., 2022, Time-lapse seismic imaging using shot gathers with non-repeatable source wavelets: Geophysics, **88**, No. 1, 1–108.

Graves, A., Jaitly, N., and Mohamed, A.-r., 2013a, Hybrid speech recognition with deep bidirectional lstm, *in* 2013 IEEE workshop on automatic speech recognition and understanding, IEEE, 273–278.

Graves, A., Mohamed, A.-r., and Hinton, G., 2013b, Speech recognition with deep recurrent neural networks, *in* 2013 IEEE international conference on acoustics, speech and signal processing, Ieee, 6645–6649.

Graves, A., and Schmidhuber, J., 2005, Framewise phoneme classification with bidirectional lstm and other neural network architectures: Neural networks, **18**, No. 5-6, 602–610.

Guo, R., Zhang, J., Liu, D., Zhang, Y., and Zhang, D., 2019, Application of bi-directional long short-term memory recurrent neural network for seismic impedance inversion, *in* 81st EAGE Conference and Exhibition 2019, vol. 2019, European Association of Geoscientists & Engineers, 1–5.

Henley, D. C., and Lawton, D. C., 2021, Time-lapse detection using raypath interferometry: Geophysics, **86**, No. 2, Q27–Q36.

Hochreiter, S., and Schmidhuber, J., 1997, Long short-term memory: Neural computation, **9**, No. 8, 1735–1780.

Hussein, M., Stewart, R. R., Sacrey, D., Johnston, D. H., and Wu, J., 2021, Unsupervised machine learning for time-lapse seismic studies and reservoir monitoring: Interpretation, **9**, No. 3, T791–T807.

Isaac, J. H., and Lawton, D. C., 2006, A case history of time-lapse 3d seismic surveys at cold lake, alberta, canada: Geophysics, **71**, No. 4, B93–B99.

Isaac, J. H., and Lawton, D. C., 2014, A case history of experimental time-lapse 3c 2d seismic reflection data for reservoir monitoring at cold lake, alberta, canada: Interpretation, **2**, No. 2, SE47–SE54.

Jordan, M., 1986, Serial order: a parallel distributed processing approach. technical report, june 1985-march 1986, Tech. rep., California Univ., San Diego, La Jolla (USA). Inst. for Cognitive Science.

Koster, K., Gabriels, P., Hartung, M., Verbeek, J., Deinum, G., and Staples, R., 2000, Time-lapse seismic surveys in the north sea and their business impact: The Leading Edge, **19**, No. 3, 286–293.

Li, D., Peng, S., Guo, Y., Lu, Y., and Cui, X., 2021, Co2 storage monitoring based on time-lapse seismic data via deep learning: International Journal of Greenhouse Gas Control, **108**, 103,336.

Li, Y., and Alkhalifah, T., 2022, Target-oriented time-lapse elastic full-waveform inversion constrained by deep learning-based prior model: IEEE Transactions on Geoscience and Remote Sensing, **60**, 1–12.

Lumley, D. E., 2001, Time-lapse seismic reservoir monitoring: Geophysics, **66**, No. 1, 50–53.

Moya, A., and Irikura, K., 2010, Inversion of a velocity model using artificial neural networks: Computers & geosciences, **36**, No. 12, 1474–1483.

Pennington, W. D., 2000, "do no harm!"–seismic petrophysical aspects of time-lapse monitoring, *in* 2000 SEG Annual Meeting, OnePetro.

Pham, N., and Wu, X., 2019, Missing sonic log prediction using convolutional long short-term memory, *in* SEG International Exposition and Annual Meeting, OnePetro.

Rickett, J., and Lumley, D., 2001, Cross-equalization data processing for time-lapse seismic reservoir monitoring: A case study from the gulf of mexico: Geophysics, **66**, No. 4, 1015–1025.

Roy, P., Zhu, X., and Fei, W., 2020, Machine learning assisted seismic inversion, *in* SEG International Exposition and Annual Meeting, OnePetro.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J., 1985, Learning internal representations by error propagation, Tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science.

Schuster, M., and Paliwal, K. K., 1997, Bidirectional recurrent neural networks: IEEE transactions on Signal Processing, **45**, No. 11, 2673–2681.

Wang, Y., and Morozov, I. B., 2020, Time-lapse acoustic impedance variations during co2 injection in weyburn oilfield, canada: Geophysics, **85**, No. 1, M1–M13.

Wang, Z., 1997, Feasibility of time-lapse seismic reservoir monitoring: The physical basis: The Leading Edge, **16**, No. 9, 1327–1330.

Wapenaar, K., and Van Ijsseldijk, J., 2021, Employing internal multiples in time-lapse seismic monitoring, using the marchenko method, *in* 82nd EAGE Annual Conference & Exhibition, vol. 2021, European Association of Geoscientists & Engineers, 1–5.

Watanabe, T., Shimizu, S., Asakawa, E., and Matsuoka, T., 2004, Differential waveform tomography for time-lapse crosswell seismic data with application to gas hydrate production monitoring, *in* SEG Technical Program Expanded Abstracts 2004, Society of Exploration Geophysicists, 2323–2326.

Yoon, D., Yeeh, Z., and Byun, J., 2020, Seismic data reconstruction using deep bidirectional long short-term memory with skip connections: IEEE Geoscience and Remote Sensing Letters, **18**, No. 7, 1298–1302.

Yuan, C., Zhang, X., Jia, X., and Zhang, J., 2020, Time-lapse velocity imaging via deep learning: Geophysical Journal International, **220**, No. 2, 1228–1241.

Zhang, R., Ghosh, R., Sen, M. K., and Srinivasan, S., 2013, Time-lapse surface seismic inversion with thin bed resolution for monitoring co2 sequestration: A case study from cranfield, mississippi: International Journal of Greenhouse Gas Control, **18**, 430–438.

Zhong, Z., Sun, A. Y., and Wu, X., 2020, Inversion of time-lapse seismic reservoir monitoring data using cyclegan: a deep learning-based approach for estimating dynamic reservoir property changes: Journal of Geophysical Research: Solid Earth, **125**, No. 3, e2019JB018,408.

Zhou, R., Huang, L., and Rutledge, J., 2010, Microseismic event location for monitoring co 2 injection using double-difference tomography: The Leading Edge, **29**, No. 2, 208–214.