PROCEEDINGS, 46th Workshop on Geothermal Reservoir Engineering
Stanford University, Stanford, California, February 15-17, 2021
SGP-TR-218

Recurrent Neural Networks for Prediction of Geothermal Reservoir Performance

Anyue Jiang, Zhen Qin, Trenton T. Cladouhos, Dave Faulder, and Behnam Jafarpour

Mork Family Department of Chemical Engineering and Materials Science, University of Southern California, Los Angeles, California, USA

anyuejia@usc.edu

Keywords: machine learning, predictive analytics, recurrent neural networks, geothermal reservoirs

ABSTRACT

Reliable prediction of energy production performance from geothermal reservoirs is needed for optimizing sustainable development of the underlying resources. While conventional physics-based simulation models offer a comprehensive prediction tool, they are nontrivial to build and involve a significant level of uncertainty due to data limitation. Data-driven models offer an efficient approach to build predictive models by extracting statistical patterns in the historical data and using them to perform direct forecasting. In particular, recurrent neural networks (RNN) are commonly used as data-driven models for the prediction of time series sequences. Using an inherent sequential architecture, RNN can learn the dynamic behavior and dependencies among different variables from data and use the learned patterns to make predictions. In this work, we investigate the application of RNN to the prediction of energy production from geothermal reservoirs. Specifically, we develop an encoder-decoder architecture, known as CNN-RNN, in which the encoder uses a convolutional neural network (CNN) to summarize the features learned from historical data while the decoder is based on a sequential RNN structure that uses those features and additional inputs to make predictions. We present the CNN-RNN architecture and demonstrate its prediction performance by applying it to different datasets, including field data from a geothermal reservoir.

1. INTRODUCTION

Improving the efficiency of energy production from geothermal reservoirs hinges on the availability of accurate prediction models to describe the performance of the geothermal reservoir under alternative operation and development scenarios. Physics-based models offer a comprehensive prediction framework that requires constructing a reliable reservoir model, which involves extensive data acquisition, integration of multiple sources of data, and simulation of complex multi-physics processes. Additionally, the uncertainty in describing reservoir models and their physical properties introduces a significant level of uncertainty in the resulting predictions. Other difficulties associated with the deployment of physics-based models include their computational burden and expertise requirement. These limitations, especially the effort required to construct a reliable model, complicate the use of physics-based simulation models for managing the operations and development planning of geothermal reservoirs.

An efficient alternative to physics-based models is data-driven approaches that extract statistical patterns and dependencies from various sources of data to develop predictive models. In recent years, deep learning has enjoyed great success in many applications, including subsurface modeling (Laloy et al., 2017; Jiang et al., 2020; Razak et al., 2020). For dynamical systems that involve data sequences, recurrent neural networks (RNN) present an effective tool for capturing the temporal trends in the data. RNN can be viewed as a directed graph with a temporal sequence that can be used to model dynamic data. An important advantage of RNN is its internal state (memory), which facilitates the extraction and use of important information from the past in generating predictions. Effective RNN models, including the Long Short-Term Memory (LSTM), proposed by Hochreiter et al. (1997), and Gated Recurrent Units (GRU), proposed by Cho et al. (2014), can learn long-term dependencies by implementing gate mechanisms to control the input, output, and update to the internal memory state. RNN has also been applied to geothermal data in recent years. Tian et al. (2017) applied RNN to an analysis of downhole gauge data. Gudmundsdottir et al. (2020) used RNN to capture well connections in geothermal reservoirs.

The goal of this study is to develop tailored RNN architectures for modeling and prediction of geothermal reservoir response using historical production and monitoring data. The developed model is based on an encoder-decoder architecture, named CNN-RNN, that consists of an encoder that implements a convolutional neural network (CNN) to extracts important features from historical data and a decoder that uses an RNN structure for prediction. The CNN encoder summarizes the dynamics within the historical data, while the decoder RNN predicts the production trends based on the learned features and the future control. We test the model by applying it to both simulated and real field data to evaluate its performance. Since field data often includes scheduled (e.g., service and maintenance periods) and unscheduled (faults and equipment failures) during which data is not available, the method should be able to handle data gaps. To enable this, we use a labeling scheme that allows the CNN-RNN model to ignore specific time steps that involve data gaps.

In this paper, we present the CNN-RNN model in detail by first introducing its architecture and the training procedure to learn the underlying network parameter. We then use different datasets, including field data from a binary cycle geothermal power plant, to demonstrate the prediction performance of the developed model.

2. METHODOLOGY

In this section, we introduce the CNN-RNN model and describe its important components. We begin by presenting a typical example of the historical dataset from a geothermal field before introducing the method.

2.1 Geothermal Data

The normalized hourly data for a producer in a geothermal reservoir is shown in Figure 1. In this case, the control variable is the pump motor speed (shown in brown) has a strong signature on other variables, which implies that the model needs to process not only historical information but also the control information. In addition, these time series generally exhibit smooth trends except for the sudden drops that are caused by the shut-in period. During these shut-in periods, the data is not considered reliable and should not be used for training. One way to handle these shut-in periods is to remove them periods, which will introduce random data gaps. Therefore, the model used to learn the behavior of this data should be able to process irregularly sampled datasets.

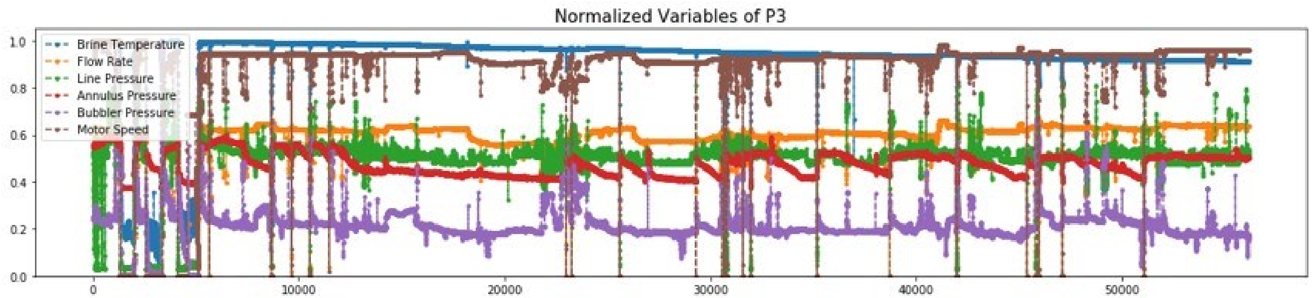
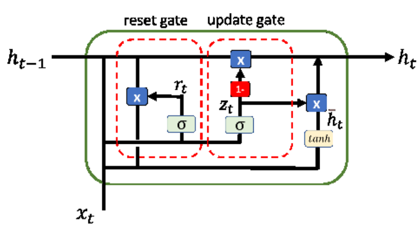


Figure 1: Normalized Hourly data of a geothermal producer.

2.2 RNN

The LSTM and GRU are two commonly used RNN models. They share the same general structure in which each unit receives input at the current time step and the hidden state from its previous unit. The major difference between the LSTM and GRU is that the former has an additional memory unit that requires more weights to control and more data to train. Since the typical size of historical data in geothermal reservoirs is relatively small, we use the GRU model in this study.

The GRU structure proposed by Cho et al. (2014) is shown in Figure 2, where each unit receives as input the hidden state from its previous unit h_{t-1} and the control at the current step x_t, and updates the hidden state h_t and generates the corresponding output. The update gate z_t decides what information to throw away and what new information to add to the state, while the reset gate r_t control how much information to forget. The variable $\tilde{h}_t$ denotes the current memory content, while is the h_t output and the hidden state of the current step.



$$r_t = \sigma(W_r x_t + U_r h_{t-1})$$
$$z_t = \sigma(W_z x_t + U_z h_{t-1})$$
$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \cdot h_{t-1}))$$
$$h_t = z_t \cdot h_{t-1} + (1 - z_t) \cdot \tilde{h}_t$$

Figure 2: A typical structure of the GRU model.

2.3 CNN

RNN is typically used with an encoder-decoder architecture, which has found popularity due to its success in machine translation. The idea is to have an encoder RNN to encode the driving series and a decoder RNN to generate predictions with the compressed information. However, an issue with RNN is that its performance deteriorates rapidly as the length of sequence increases, but the model needs to extract information within a long history in time series prediction. A common solution is to use the attention mechanism proposed by Bahdanau et al. (2014) to select important parts in the driving series. Geothermal data is typically smooth, except for the shut-in periods. Therefore, an easier alternative is to replace RNN with a one-dimensional CNN to extract the main features and correlation patterns in the historical data. An illustration of 1D convolution is shown in Figure 3, where w_1, w_2, w_3 are the weights in the convolutional filter. Compared to RNN, CNN has fewer weights and is more efficient when dealing with long sequences.

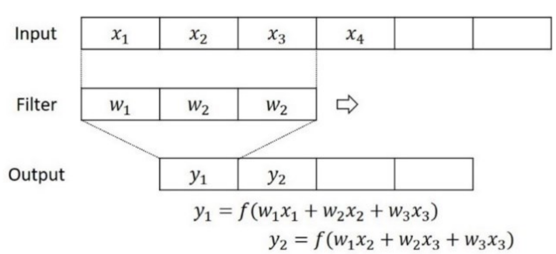


Figure 3: 1D convolution.

2.4 Labeling Scheme

As stated in Section 2.1, after removing the shut-in periods, the resulting data gaps lead to an irregularly sampled dataset. One way to handle this type of data is to fill the gaps through interpolation. Interpolation schemes, however, incorporate assumptions and introduce bias. An alternative approach, proposed by Zipton et al. (2016), is to label the missing data and let RNN learn to make predictions with the existing data gaps. We apply this labeling scheme to the data and learning algorithm. To implement this approach, first, a label vector is generated by assigning a label (0) to data points during the shut-in periods that must be ignored, while other data points are given a different label (1). Accordingly, the observations during the shut-in period are also replaced with zeros. In the CNN-RNN model, the label vector is included as part of the control to inform the model which steps should be ignored. In addition, during those steps, the output of the model is multiplied by the provided label to get consistent outputs during training. The mechanism essentially excludes the data during shut-in periods from the loss function.

2.5 CNN-RNN Model

The overall structure of the CNN-RNN model is shown in Figure 4, where t, x, y, $\hat{y}'$, $\hat{y}$, L are the time step, control input, observation, prediction of state, final prediction, and the label, respectively. The model combines the CNN's efficiency in summarizing information and the RNN's ability to learn and predict dynamic (sequence) data. The CNN encoder efficiently summarizes the useful information in the historical data and provides the learned feature vector to the decoder RNN. The decoder RNN predicts the target sequence by combining the control information and the compressed feature vector. The final prediction is the output from RNN multiplied by the label. Although the RNN model works with fixed-length sequences, it can predict arbitrary long future with the feed-forward scheme where predictions are reused as history.

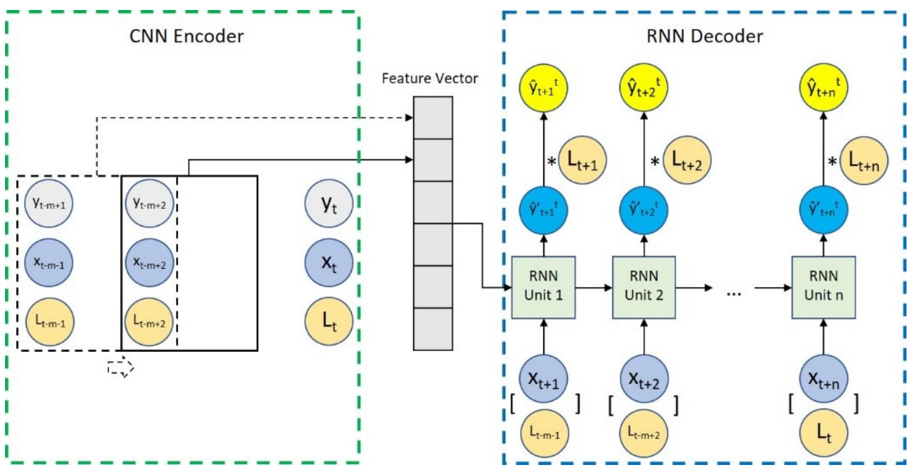


Figure 4: The CNN-RNN model structure.

3. NUMERICAL EXPERIMENT

To evaluate the performance of the CNN-RNN model, we use it in a simple synthetic example before applying it to a field dataset. In these examples, we evaluate the performance of the model for single-window prediction and multi-window (long-term) predictions using a feed-forward scheme. The numerical examples in this study are implemented in Tensorflow (Ababi et al., 2016).

3.1 Synthetic Dataset

In our first example, we design five experiments using a periodic cosine function to test different aspects of the CNN-RNN model. The five experiments are shown in Figure 5. Test 1 is the base case, where the data has constant frequency and amplitude. Test 2 has an increasing frequency and amplitude. Test 3 contains variable frequency and amplitude, first increasing but then decreasing in the latter part. Test 4 includes data corrupted with random noise. Finally, Test 5 is designed to test the model performance under the data gap. In this case, 10% of the data are randomly selected and labeled as 0. Each test has a total of 498 steps, where the last 96 steps are reserved as the prediction period (test dataset). The predictions are made for three steps ahead (one window) by using the data in the last 24 steps (as history). The performance of the CNN-RNN model is compared with those of CNN and RNN with a similar level of complexity. For each test, the models are trained 20 times to get the average performance.

The results of single window prediction are shown in Figure 6, where the vertical dashed lines separate the training and test sets. The models show good performance in the tests except in Test 5. The RNN and CNN models do not show good performance when data gaps are present, whereas the CNN-RNN model is not affected by the data gaps because of the labeling scheme. For multi-window predictions, all prediction steps are predicted at once at the last training step without updating the models with new observations. From Figure 7, the models provide good predictions when the trend in data (amplitude and frequency) does not change, even when noise is present. However, when the trend in the data varies, the models cannot predict the trend correctly. This is, in part, because the historical data does not include repeated patterns. In other experiments (not shown), when a longer historical dataset with repeated patterns (with variable frequency and amplitudes) were used, the methods showed a better performance.

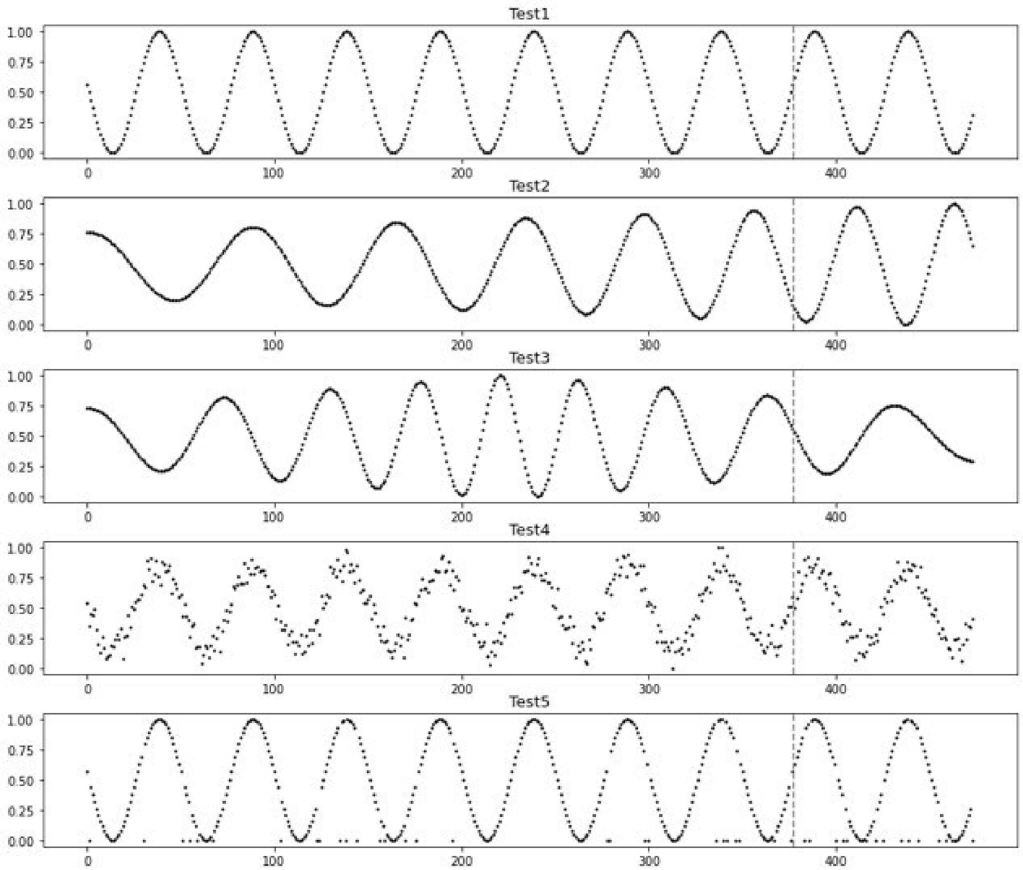


Figure 5: Data from the simple synthetic tests.

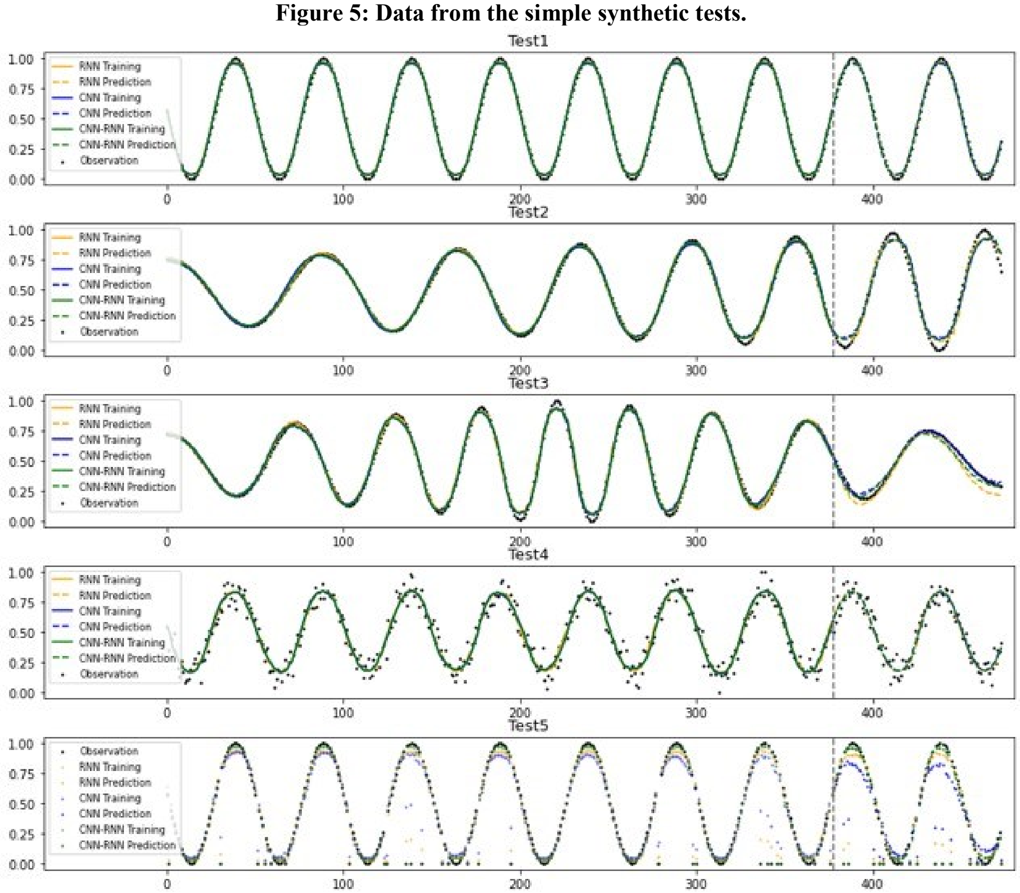


Figure 6: Average single-window predictions.

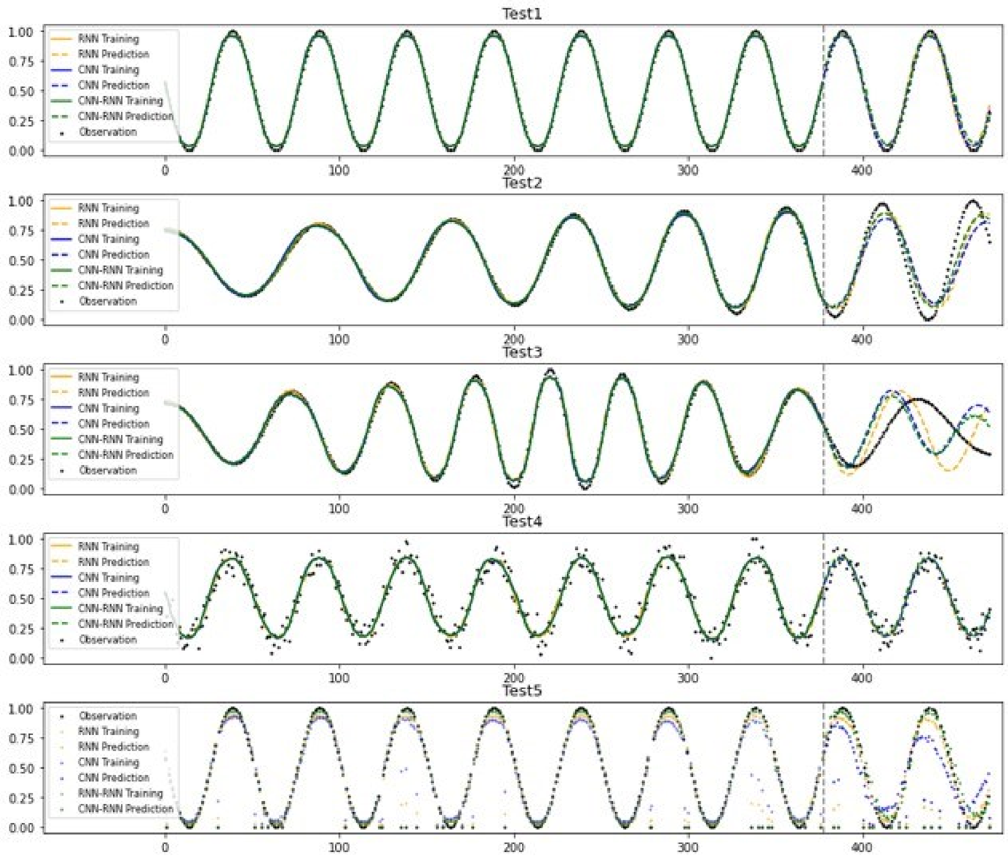


Figure 7: Average multi-window predictions.

To summarize the results from our first example, Figure 8 shows the prediction errors. Overall, the RNN and CNN-RNN models have a better prediction performance than that of CNN. In these examples, the RNN and CNN models do not have the labeling scheme, which degrades their performance. The labeling scheme makes the CNN-RNN model robust to the presence of data gaps. We note that results in this example and the plot in Figure 8 are based on models with similar complexity for comparison purposes. Allowing each model to have its own distinct architecture may lead to performance improvement for each case.

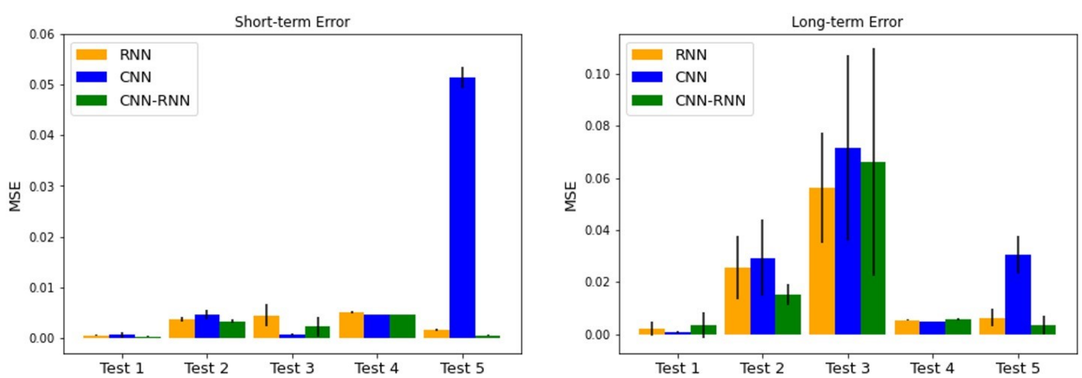


Figure 8: Summary of errors in the synthetic tests.

3.2 Field Dataset

The field dataset has hourly data with a total of 51568 data points. Each data point has the average value of a variable over one hour (the raw data in the field is measured every minute). In our example, the last 8640 timesteps are used for prediction, and the rest are separated into training and validation sets (90% training, 10% validation). In this example, we first identified the relevant variables and grouped them as controls and observations (Table 1). For the injectors, although the wells are controlled by the motor speed of the pumps, we assigned the flow rate of the injectors as a replacement since the motor speed of the injectors is not available. Another control variable for the injectors is the brine temperature, which is not controlled but is provided as the output from the powerplant. For the producers, since the motor speed is available, every other variable is categorized as an observation. For each RNN sequence, data from the past 48 hours are used as history, and the prediction window is 24 hours. These lengths are some of the hyperparameters that are tuned during training.

Table 1: The list of control (C) and Observed (O) variables.

	Injector			Producer					
Variable	Line Pressure	Flow Rate	Brine Temp	Line Pressure	Flow Rate	Brine Temp	Motor Speed	Bubbler Pressure	Annulus Pressure
Unit	PSIG	GPM	F	PSIG	GPM	F	RPM	PSIG	PSIG
Control / Observation	O	C	C	O	O	O	C	O	O

The predicted brine temperature and flow rates are shown in Figures 9 and 10, where the vertical line separates the training and test parts. The single window (short-term) predictions provide a good match to the true observations. The multi-window predictions are not as accurate due to error accumulation. An important factor that can affect the performance of multi-step predictions is the possible changeS in data trends during the prediction window. In addition, the model is not updated with incoming data, a factor that can be used to improve the prediction performance. Overall, the performance of the multi-step prediction is as expected, considering the sudden changes that are introduced and the error accumulation issues.

We also removed the shut-in periods based on the control information to consider the main trend in the data without the short-lived transient effects. When the control is significantly smaller than its normal range, the target variables are set to zero, and a zero label is assigned to these steps. The results are shown in Figures 11 and 12. We note that this simple labeling scheme does not remove all the effects that are introduced due to shut-in periods because some short-term fluctuations result after each shut-in period. Since these variations do not have a related signature in the control vector, they are not likely to be predictable by the model. However, additional investigation is needed to develop more sophisticated techniques to deal with the shut-in period and to examine the predictive capability of the model.

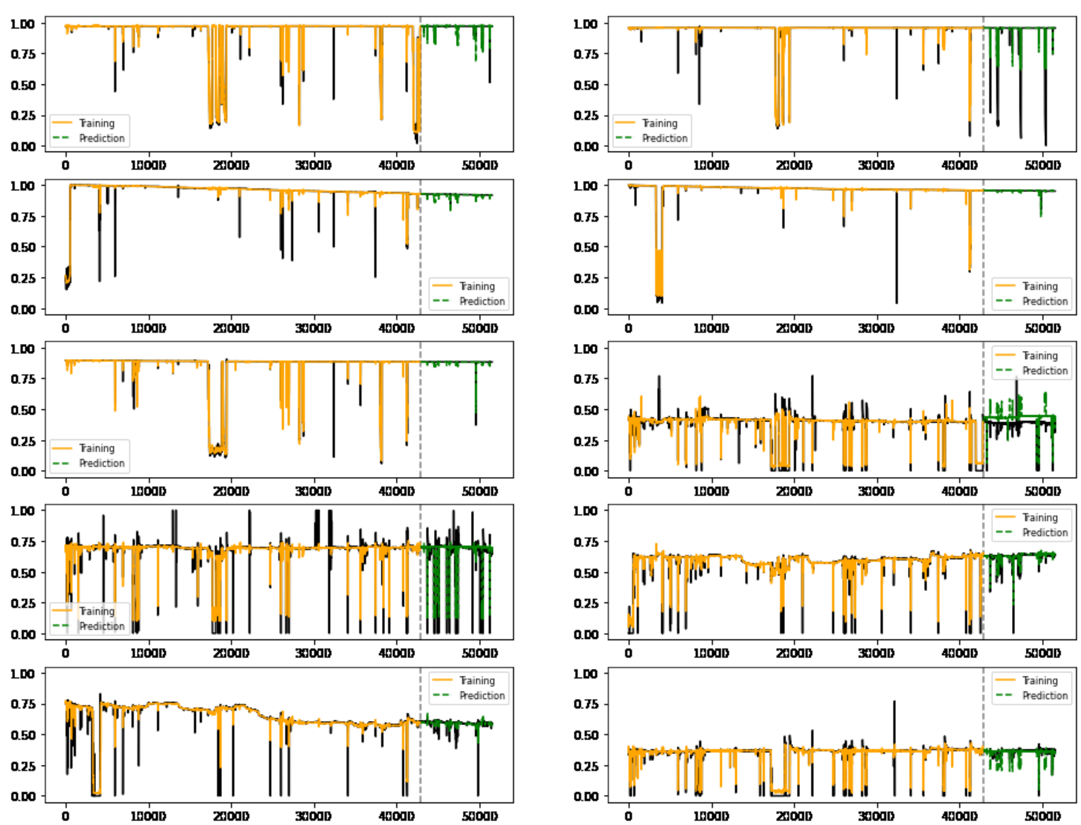


Figure 9: Single window predictions for field data.

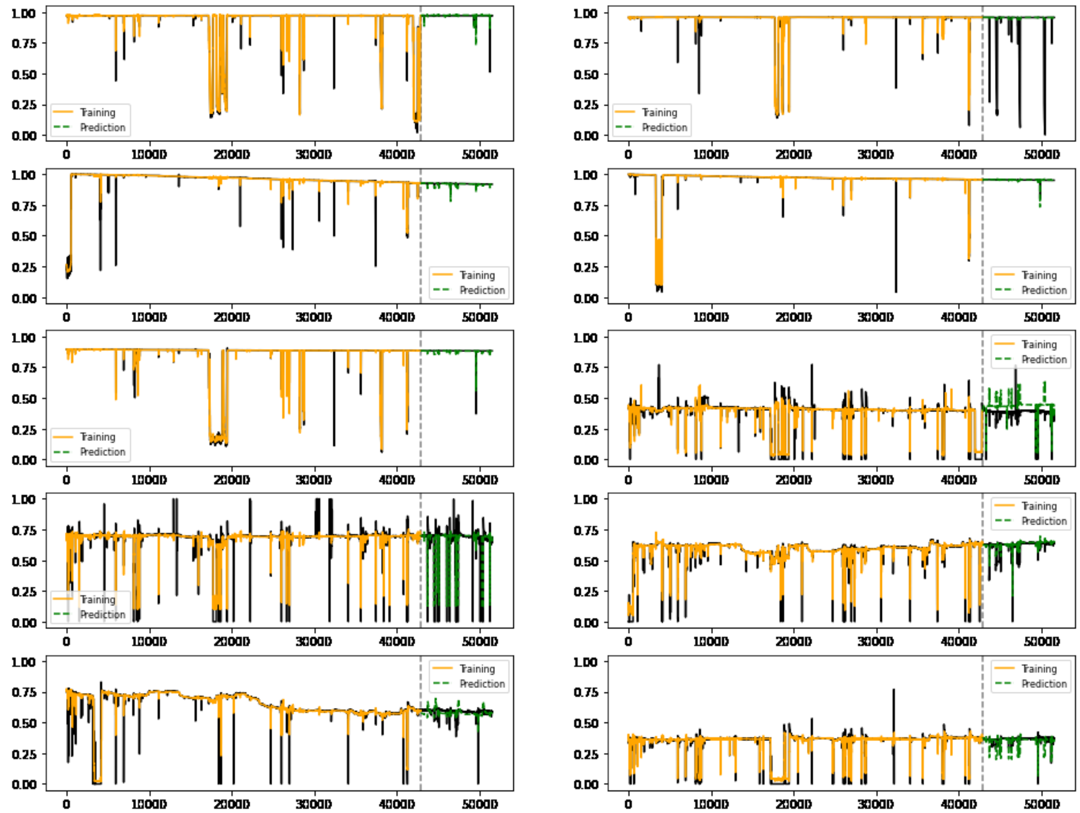


Figure 20: Multi-window predictions results for field data.

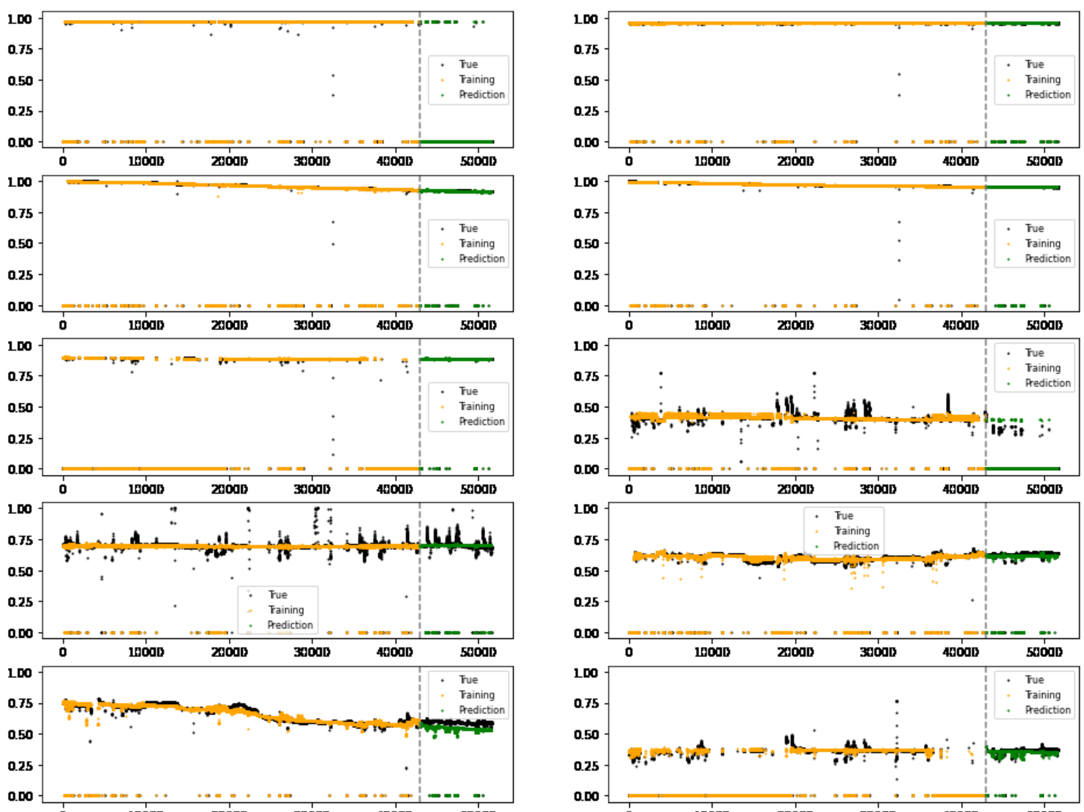


Figure 31: Single-window predictions results with simple labeling scheme.

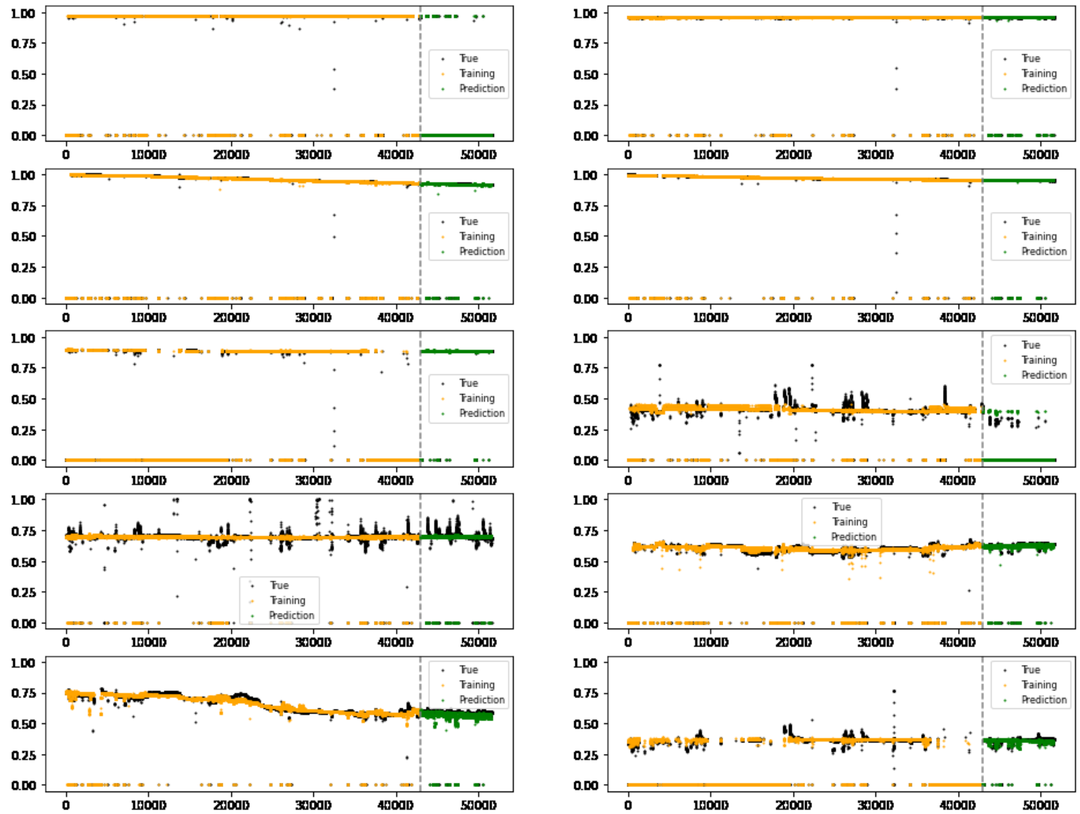


Figure 42: Multi-window predictions with the simple labeling scheme.

4. DISCUSSION AND CONCLUSION

In this study, we have developed a robust CNN-RNN model for the analysis and prediction of geothermal energy production data. The model combines the efficiency of CNN in summarizing the features in long data sequences with the predicting power and properties of RNN. An additional feature of the model is its ability to handle irregularly sampled data using a simple labeling scheme to avoid biased introduced through interpolation schemes. Using synthetic examples and a field dataset, we evaluated the performance of the developed model and compared its performance with standard CNN and RNN models. While the model shows good performance for single-step and multi-step predictions, multi-step prediction results are not affected by error accumulation. Another important consideration to improve the performance of the model is retaining based on incoming data during the prediction phase, which was not performed in this work. Retraining is helpful when the trends observed during the prediction are different from those seen during the training stage (historical data). Additional investigation is also needed to improve the labeling scheme and to develop RNN models for long-term prediction (over the years) and comparison with simulation-based predictions.

ACKNOWLEDGEMENT

This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Geothermal Technologies Office award number DE-EE0008765. Numerical experiments and codes can be found at https://github.com/AnyueJ/Geothermal_Prediction.

REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. In 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16) (pp. 265-283).

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Gudmundsdottir, H., & Horne, R. N. Prediction Modeling for Geothermal Reservoirs Using Deep Learning, PROCEEDINGS of the 45th

Workshop on Geothermal Reservoir Engineering, Stanford University, Stanford, California, February 10-12, 2020. SGP-TR-216.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

Jiang, A., & Jafarpour, B. (2020, September). History Matching under Uncertain Geologic Scenarios with Variational Autoencoders. In ECMOR XVII (Vol. 2020, No. 1, pp. 1-14). European Association of Geoscientists & Engineers.

Jiang, A., & Jafarpour, B. (2020). Inverting Subsurface Flow Data for Geologic Scenarios Selection with Convolutional Neural Networks. Advances in Water Resources, 103840.

Laloy, E., Hérault, R., Lee, J., Jacques, D., & Linde, N. (2017). Inversion using a new low-dimensional representation of complex binary geological media based on a deep neural network. Advances in Water Resources, 110, 387-405.

Razak, S. M., & Jafarpour, B. (2020). Convolutional neural networks (CNN) for feature-based model calibration under uncertain geologic scenarios. Computational Geosciences, 24(4), 1625-1649.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. Advances in neural information processing systems, 27, 3104-3112.

Tian, C., & Horne, R. N. (2017, October). Recurrent neural networks for permanent downhole gauge data analysis. In SPE Annual Technical Conference and Exhibition. Society of Petroleum Engineers.

Lipton, Z. C., Kale, D. C., & Wetzel, R. (2016). Modeling missing data in clinical time series with rnns. arXiv preprint arXiv:1606.04130.