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

Comparative Study of Decline Curve Prediction in Geothermal Injection Well Using Machine Learning and Wellbore Simulator

Muchamad Harry[1], Muhammad Aditya F. Wahyudi[2], Muhammad Midat P. Al Islam[3], Nada A. Sabrina[2], and Jantiur Situmorang[1]

[1]AILIMA, Jakarta, Indonesia; [2]Bandung Institute of Technology, Bandung, Indonesia; [3]Polytechnic of Energy and Mineral Akamigas, Cepu, Indonesia

harry@ailima.com, muhammadadityafw@gmail.com, midat.07@gmail.com, nadasabrina99@gmail.com, jantiur@ailima.co.id

Keywords: injection, injection decline, machine learning, JIWA Flow

ABSTRACT

Reinjection plays a crucial role in geothermal field operation and management. The sustainability of geothermal power plants depends on how field injection capacities can be maintained, and the reinjection can effectively provide pressure support to producer wells. A good monitoring of injection performance is therefore essential in achieving excellent field management. Currently, analyzing and monitoring injection well issues & performances generally rely on wellbore simulation, which is challenging and very time-consuming. This study aims to develop an easier and faster alternative solution for analyzing geothermal injection well's performance by applying machine learning (ML). The ML model was compiled, trained, and validated using synthetic data generated from a wellbore simulator (JIWA Flow). This study's best algorithm is Feed Forward Neural Network (FFNN) with R Squared and MSE of 0.998 and 3.4. It is proved by the comparison between the unseen data from JIWA Flow and prediction from FFNN. The application of the proposed ML to calculate the exponential annual injection decline from several scenarios are provided.

1. INTRODUCTION

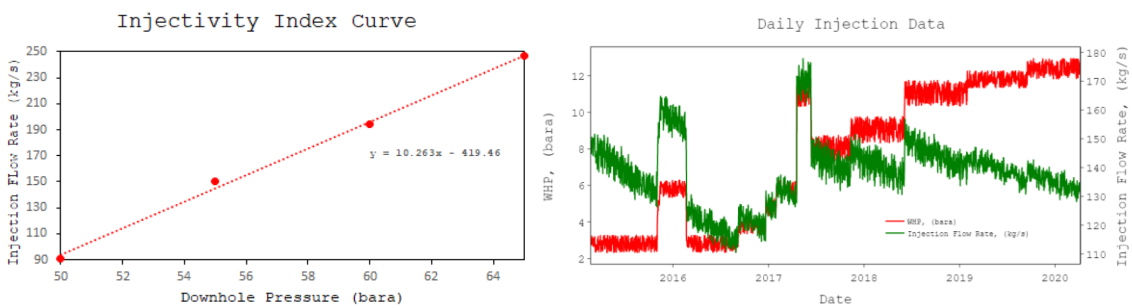
Reinjection during geothermal field operation is intended to serve two purposes: improve resource recovery and discharge wastewater. Careful injection system design is required to achieve these two purposes and provide good field management. A good understanding of injection practices is essential for optimal development and management of geothermal resources. During field management, several challenges to maintain reinjection capacity might be encountered, such as a limited reservoir, scaling in the wellbore, and permeability reduction. Modeling using conventional wellbore simulators is required to solve this problem. However, this process consumes a lot of time because of complex data preparations and many manual iterations required in the wellbore modeling. This paper is attributed to applying a machine learning method as a potential predictor for injection mass flow rate in a faster manner. Comprehensive validation was carried out by comparing the program output and injection mass flow rate from the wellbore simulation. Some of the objectives that will be achieved through this study are determining which machine learning methods give the best match with the dataset constructed from a wellbore simulator and applying the proposed machine learning techniques under various problems that may occur in the injection well.

2. LITERATURE REVIEW

Reducing greenhouse gas emissions was a highlighted goal in the debate on Global Warming / Climate Change and the recent World Environment Summit (Kyoto Protocol). Geothermal energy, one of the renewable energy sources, is being preferred because it is sustainable and environmentally friendly. It is necessary to have a sustainable understanding of reservoir management strategies to increase sustainable development from alternative energy sources. The core of far-sighted geothermal reservoir engineering and management is to achieve sustainable development of an inexhaustible resource, meet reservoir longevity / environmental protection concerns, and economically competitive standards. Thus, an integrated approach to a sustainable reservoir management strategy can be described in **Figure 1**.

Figure 1: Reservoir management diagram (Ungemach and Antics, 2003)

Geothermal injection plays a key role in managing geothermal resources for several purposes, such as providing pressure support, improving thermal recovery, enhancing or revitalizing surface thermal features, etc. Within the maintainable utilization of geothermal assets, there are a few issues related to injection, such as cooling of production wells, silica scaling in surface pipelines and injection wells, and rapid clogging of aquifers nearly injection wells sandstone reservoirs (Axelsson, G., 2012). This monitoring can be in the form of a measure of well fluid uptake at a certain WHP or a downhole pressure called the injectivity index (Grant and Bixley, 2011). The fitting of the well injectivity usually is one straight line to get the average well injectivity, as shown in **Figure 2.a.** However, with a large amount of injection flow rate and WHP data shown in **Figure 2.b.**, it will take a relatively long time to analyze and monitor injection well problems using numerical models.



(a) (b)

Figure 2: (a) Typical Injectivity Index curve, (b) Typical daily injection data

There have been studies to apply machine learning in geothermal industries. From forecasting flow rates of production wells (Ariturk, 2019) to the estimation of power generated from the turbine and the grid (Aldi et al., 2019). There are so many types of machine learning out there, and it is not a guarantee that a complex architecture of machine learning will bring out better accuracy in the case of prediction. In this study, we use a regression method to build our model. Regression is a process of estimating a target value from a set of features. This method is mostly used for forecasting and finding out cause and effect relationships between variables. Regression techniques mostly differ based on the number of independent variables and the relationship between the independent and dependent variables. There are several types of algorithms to solve the regression equation.

3. METHODOLOGY

The workflow of this study is shown in **Figure 3**. The process was started by doing data generation to be used as the database for model machine learning. The data is in the form of a date, wellhead pressure, and injection mass flow rate generated from JIWA Flow wellbore simulator. Next, history matching was conducted to mimic the dataset. History matching consists of model training and model evaluation. Model training is developed to get the model to be tested in different terms. Split the datasets into two: training dataset, and testing data set, are required. Then, model evaluation is to choose a model that fits the testing dataset and make predictions to check the model's suitability to the dataset. To calculate injection decline, the injection data is normalized at a constant WHP. Normalization results from machine learning will be checked using normalization from a wellbore simulator. The exponential decline from normalized injection data was also calculated. Finally, the study's objectives are solved through analysis and discussion of the result.

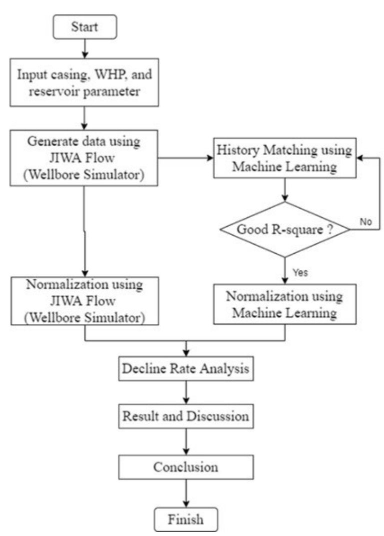


Figure 3: Workflow of this study

4. CASE STUDY & DATA GENERATION

Several types of algorithms have been used in this study. There are four datasets generated using JIWA Flow wellbore simulator to represent different injection well conditions, including: (1) no issue, (2) increase of reservoir pressure due to limited fracture volume, (3) decrease of Injectivity Index, and (4) decrease of Injectivity Index and wellbore scaling. The details of the case are shown in **Table 1** and **Figure 4**.

Table 1: Case Study

Case	Reservoir Pressure	Injectivity Index	Wellbore Scaling
1	Constant	Constant	No
2	Incline	Constant	No
3	Constant	Decline	No
4	Constant	Decline	Occur

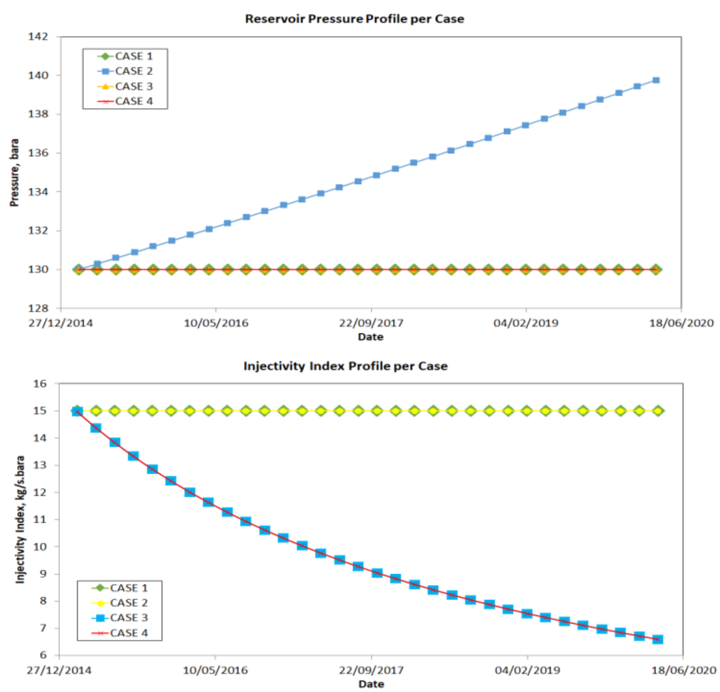


Figure 4: Input Pressures and Injectivity Indices for Case 1, 2, 3, and 4

The synthetic data of WHP and injection mass flow rate generated from 23 February 2015 to 6 April 2020 using JIWA Flow wellbore simulator. The fluid injection temperature is around 120°C. The feed zone is assumed at 1500 m with an initial pressure of 130 bar and an initial Injectivity Index of 15 kg/s.bar. Big hole casing design is used as provided in **Figure 5**. As a result of the wellbore simulation using the above assumption, generated WHP and injection flow rates are provided in **Figure 6**.

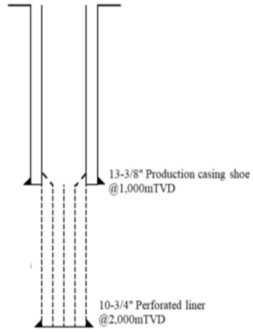


Figure 5: Well diagram in wellbore simulator

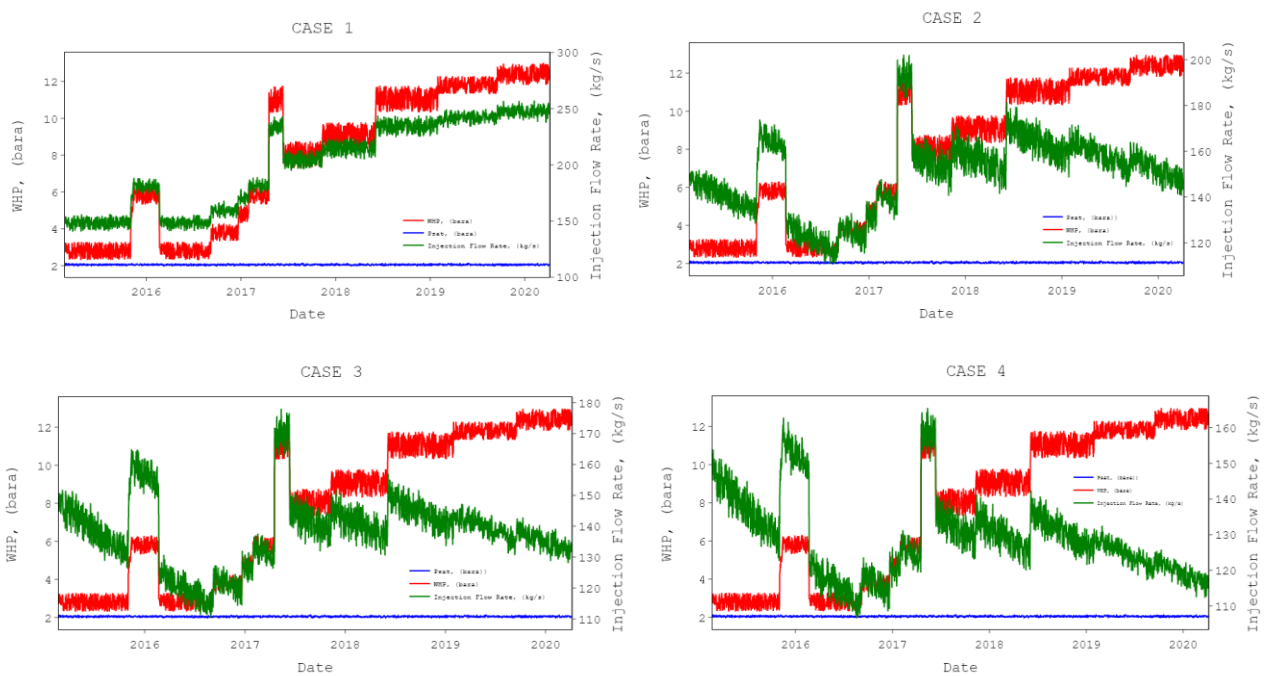


Figure 6: Synthetic data generated from JIWA Flow wellbore simulator

5. MODEL TRAINING

For the first step, data generated from case 1 was trained using several machine learning algorithms. The data is split into 80:20 as a training and testing dataset, respectively. To figure out the best hyperparameter used in the training process, a function is utilized to loop through predefined hyperparameters while fitting the model to the training dataset. Therefore, the best parameters can be selected depending on the results at each looping. In this study, several machine learning regression models have been tested, which can be seen in (see Appendix A). To optimize the training process model, hyperparameter tunings were performed for each machine learning regression model, as shown in **Table 2**.

Table 2: Tuned Hyperparameter

Model	Tuned Hyperparameter
Support Vector Regression (SVR)	gamma, C
K Nearest Neighbor Regression (KNN)	n_neighbor
Random Forest Regression	max_depth, n_estimators
Gaussian Process Regression (GPR)	kernel, n restart_optimizer

Model	Tuned Hyperparameter
Feed Forward Neural Network (FFNN)	neuron, activation functions
Decision Tree Regressor	splitter, criterion, max_features
Stochastic Gradient Descent (SGD)	tol
Lasso Regression	tol, alpha
Ridge Regression	tol, alpha
General Regression Neural Network (GRNN)	kernel, sigma, calibration

6. MODEL EVALUATION

It is essential to define the model's accuracy in predicting the new dataset compared to the actual value. In this study, the models were evaluated by their Mean Absolute Error (MAE), Mean Squared Error (MSE), and R Squared functions that were automatically issued by our program interface. Another evaluation method is based on the data visualization between prediction on the data test vs. actual data test. Therefore, we can determine which model succeeded in making predictions that match the actual data. The model training results indicate that all the algorithms succeeded in achieving a high value of R Squared, as shown in **Table 3**. Qualitatively, the data visualization between the prediction data test against the actual test data for all models also shows promising results, the more detailed in **Figure 7**.

Table 3: R Squared comparison prediction

Model	MAE	MSE	R Squared
K Nearest Neighbor Regression (KNN)	1.652	4.123	0.997
Gaussian Process Regression (GPR)	1.570	3.542	0.998
Support Vector Regression (SVR)	3.695	20.031	0.986
Random Forest Regression	1.684	4.240	0.997
Feed Forward Neural Network (FFNN)	1.575	3.558	0.998
Decision Tree Regressor	1.992	6.395	0.995
Stochastic Gradient Descent (SGD)	1.642	4.074	0.997
Lasso Regression	1.633	4.026	0.998
Ridge Regression	1.633	4.026	0.998
General Regression Neural Network (GRNN)	1.556	3.588	0.998

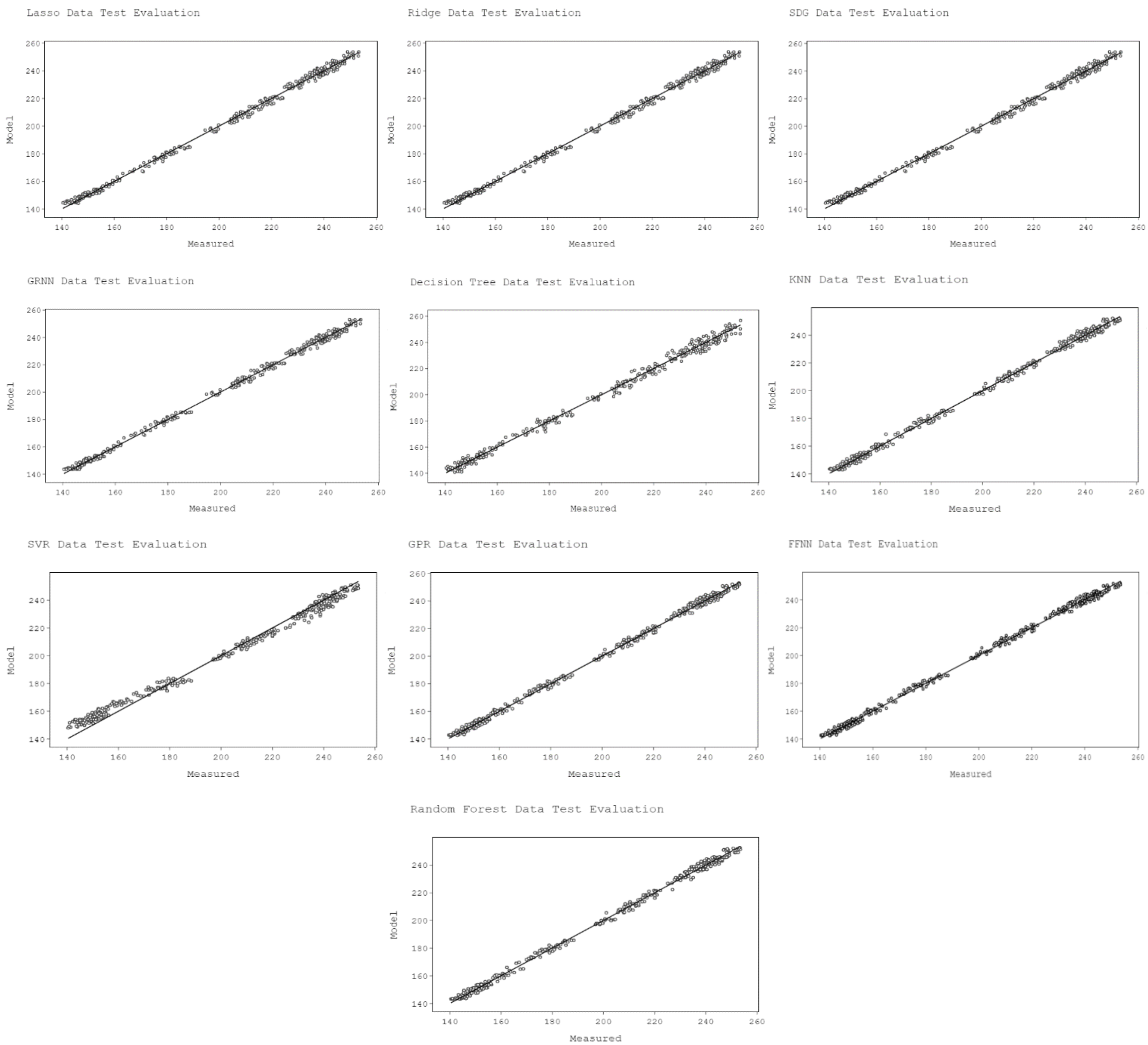


Figure 7: Data test evaluation for all machine learning models

Afterward, since R Squared's difference is relatively small, we eliminated machine learning methods based on the accuracy represented by MSE rather than R Squared during the history matching stage. The smaller the MSE, the closer the fit is to the data. So, we selected three machine learning methods: Feed Forward Neural Network Regression, Gaussian Process Regressor, and General Regression Neural Network for normalizing the production data at certain WHP.

7. RESULT AND DISCUSSION

As the results obtained in the previous evaluation, normalization have been performed for three machine learning algorithms to forecast injection flow rate at a constant WHP all the time. In this study, we use the median WHP value, which is 8.55 bara. Furthermore, a normalization curve is generated with the date as the x-axis and the injection mass flow rate as the y-axis. Then, the normalization curve is compared with the normalization results using JIWA Flow. It aims to determine the best model that produces the normalized results that are as close as possible to the normalized JIWA Flow data. The visualization shows that the Feed Forward Neural Network (FFNN) and General Regression Neural Network are comparable with the wellbore simulator, as shown in **Figure 8**. Although the three algorithms we have tried are quite excellent in model training, there are no guarantees that the model will result in the same performance as earlier while dealing with the unseen data.

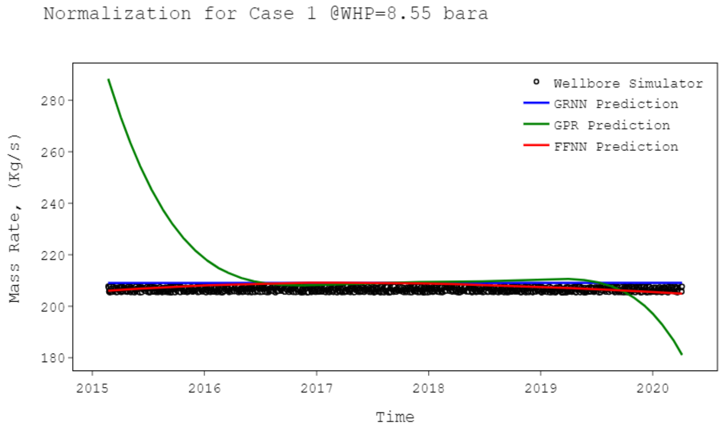


Figure 8: Normalized injection flow rate comparison between wellbore simulator, GRNN, GPR, and FFNN of case 1.

Then, decline curve analysis is performed to determine the percentage reduction in injection mass flow rate per year. The estimated decline rate per year for FFNN, GPR, GRNN, and wellbore data are 0%, 3%, 0%, and 0%, respectively. In case 1, which is in normal condition, the best machine learning methods are FFNN and GRNN regressors. Furthermore, the other three cases were tested to ensure the model's robustness. Since FFNN and GRNN are the only models that perform better in the first case, we only consider these two models for the other case. Using the same workflow processed earlier, the error from model validation for each case is quite good with R Squared more than 0.9, shown in **Table 4**. It indicates the model can be used in a various dataset that represents the different well injection problem.

Table 4: Error validation for all cases

Error Validation	FFNN				GRNN			
	Case 1	**Case 2**	**Case 3**	**Case 4**	**Case 1**	**Case 2**	**Case 3**	**Case 4**
R Squared	0.998	0.992	0.992	0.988	0.998	0.991	0.984	0.988
MAE	1.6	1.2	1.0	1.1	1.6	1.4	1.2	1.1
MSE	3.6	2.3	1.3	1.6	3.6	2.8	2.2	1.7

Normalization results of cases 2,3, and 4 is provided in **Figure 9**. The FFNN results are comparable with the wellbore simulator. Meanwhile, the GRNN poorly matches the normalized data. Even though the FFNN still cannot perfectly match the wellbore simulator's decline rate, the difference is relatively small, and thus it can be neglected. The calculated decline rate per year can be seen in **Table 5**. FFNN has significantly better performance to predict the annual decline rate.

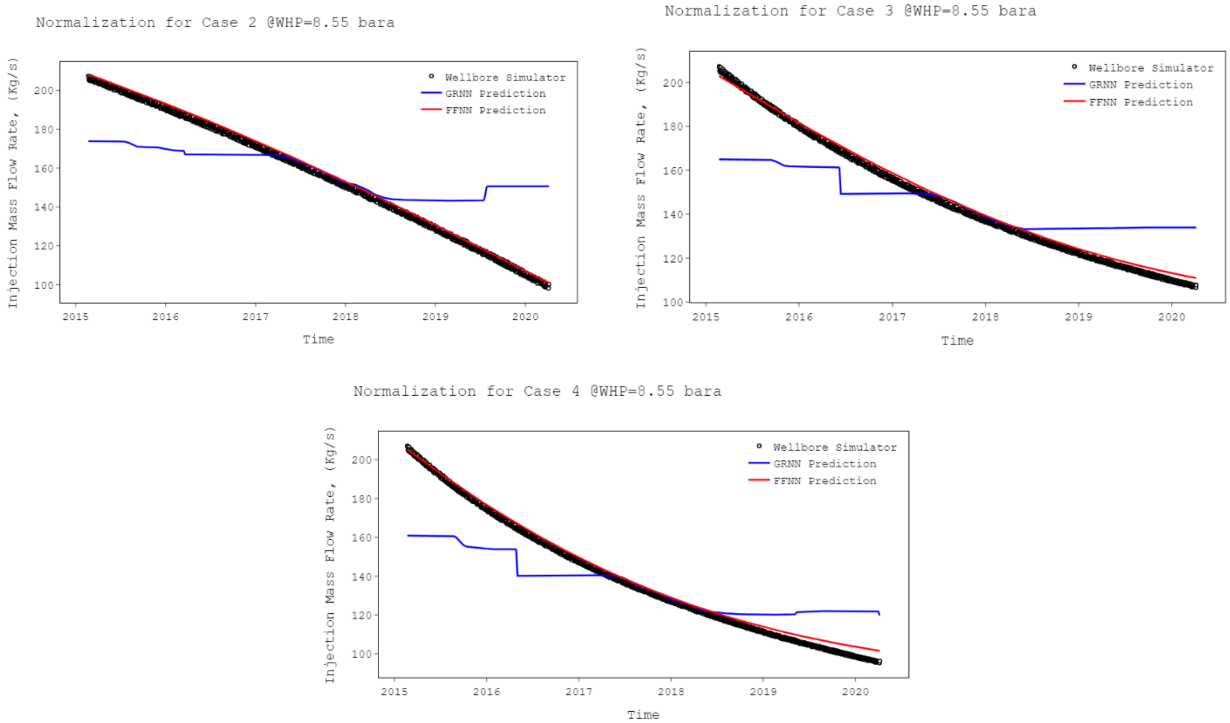


Figure 9: Normalized injection flow rate comparison between wellbore simulator, GRNN, GPR, and FFNN for each case

Table 5: Annual decline rate of injection well

Model	**JIWA Flow (%/year)**	**FFNN (%/year)**	**GRNN (%/year)**	**Error FFNN (%)**	**Error GRNN (%)**
Case 1	0	0	0	-	-
Case 2	13.9	13.9	4.3	0.1	69.4
Case 3	17.7	17.5	3.8	1.4	78.8
Case 4	14.8	14.1	6.8	4.4	53.8

8. CONCLUSION

Below are the conclusions that can be drawn from this study:

1. A model that exhibits an excellent statistical value in the training process not necessarily will give a good result while working with the unseen data.
2. From the ten machine learning methods tested in this study, the best results were obtained from Feed Forward Neural Network. It is shown that the normalized injection flow rate results are comparable with the wellbore simulator. All the different problems in injection wells, as represented by different data sets, can be reasonably matched by the model with an error of less than 5%.

ACKNOWLEDGEMENT

This proceeding was made with support and supervision of PT. Anugerah Indonesia Lima (AILIMA) through AILIMA GROW program. The resource model was simulated using JIWA Flow by PT. Anugerah Indonesia Lima (AILIMA).

REFERENCES

Ariturk, Muhammet Salih.: Optimizing the Production and Injection Wells Flow Rates in Geothermal Field Using Artificial Intelligence, Graduate Theses, Dissertations, and Problem Reports, 3772 (2019).

Axelsson, G.: Management of geothermal resources, *Proceedings,* Workshop for Decision Makers on the Direct Heating Use of Geothermal Resources in Asia, UNU-GTP, TBLRREM and TBGMED, Tianjin, China, (2008b), 15 pp.

Axelsson, G.: Role and Management of Geothermal Reinjection, UNU-GTP and La-Geo, El Savador (2012).

Bromley, C.J., Mongillo, M., and Rybach, L.: Sustainable utilization strategies and promotion of beneficial environmental effects – Having your cake and eating it too, *Proceedings,* New Zealand Geothermal Workshop 2006, Auckland, New Zealand, (2006), 5 pp.

Gudmundsdottir, Halldora. & Horne, Roland N. (2020). Prediction Modelling for Geothermal Reservoirs Using Deep Learning. Proceedings, 45th Workshop on Geothermal Reservoir Engineering Stanford University, Stanford, California, February 10-12, 2020 SGP-TR-216.

Specht, D.F.: A general regression neural network, *IEEE transactions on neural networks*, *2*(6), (1991), pp.568-576.

Appendix A
Regression Algorithm Comparison

Algorithm	Description	Pros	Cons
Decision Tree	Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output.	- Fast for inference	- Tends to overfit - Inflexible
Stochastic Gradient Descent	Stochastic in nature i.e. it picks up a “random” instance of training data at each step and then computes the gradient making it much faster as there is much fewer data to manipulate at a single time,	- Efficient. - Ease of implementation	- Requires a number of hyperparameters - Sensitive to feature scaling
Lasso Regression	A regression analysis method that performs both variable selection and regularization. Lasso regression uses soft thresholding. Lasso regression selects only a subset of the provided covariates for use in the final model.	- Rarely overfitting	- Model selected by lasso is not stable - Worse MSE
Ridge Regression	A technique for analyzing multiple regression data. When multicollinearity occurs, least squares estimates are unbiased. A degree of bias is added to the regression estimates, and a result, ridge regression reduces the standard errors.	- Prevents overfitting - Simple - Computational Efficient	- Leads to dimensionality reduction: a high bias error
General Regression Neural Network (GRNN)	A single-pass neural network which uses a Gaussian activation function in the hidden layer. GRNN consists of input, hidden, summation, and division layers.	- Single-pass learning so no backpropagation is required. - High accuracy in the estimation since it uses Gaussian functions. - Can handle noises in the inputs. - Requires only a smaller number of datasets.	- Huge size, which would make it computationally expensive. - No optimal method to improve it.
K Nearest Neighbor Regression	A regression technique that uses feature similarity to predict the values of any new data points	- No training steps - Easy and simple algorithm - Nonparametric algorithm	- Slow predictions as the number of independent variables increase - Very sensitive to outlier
Random Forest Regression	A regression technique that aggregates multiple decision trees and uses averaging to improve the accuracy and avoid over-fitting	- Handle missing values effectively - Suitable for large dataset - Can handle data with higher dimensionality	- Tend to overfit while performing with regression that have particularly noise data
Support Vector Regression	Regression technique that implements support vector mechanism, this mechanism gives us flexibility to define a decision boundary and find the best hyperplane to fit the data.	- Good performance in high dimension - Kernel flexibility	- Not suitable for large dataset - Poor performance when the dataset has more noise
Gaussian Process Regression	A nonparametric, Bayesian approach to regression, GPR calculates the probability distribution over all admissible functions that fit the data	- Directly expose the model uncertainty	- Poor performance on huge dataset
Feed Forward Neural Network Regression	A feedforward neural network that generates a set of outputs from a set of inputs	- Can easily interpret a nonlinear relationship between independent and dependent variables - Work best with huge dataset - Can be developed using multiple different training algorithm	- A 'black box' that limit us to explicitly identify possible causal relationships - Require a powerful hardware system