Exercise: Identification of Pollution Concentrations in a Shallow Groundwater Aquifer using Data-Driven Approaches

Roger Roca Miró
Legacy Contamination and Soil Remediation
TU Bergakademie Freiberg

March 11, 2025

1 Introduction

Groundwater contamination is a critical issue with significant environmental and public health implications. Accurate prediction of pollutant concentrations is essential for effective remediation and risk management. In this report, we apply the methodology of Taherdangkoo et al. [8] by adapting feedforward neural network (FFNN) training algorithms—Levenberg–Marquardt (LM) and Bayesian Regularization (BR).

The LM algorithm, introduced by Hagan and Menhaj (1994), is recognized for its computational efficiency and rapid convergence when training FFNN. MATLAB implements this algorithm in the `trainlm` function, which has become one of the most widely used tools for training neural networks due to its reliability and speed [3, 4].

The Bayesian Regularization algorithm, detailed by Foresee and Hagan (1997) [1, 4], enhances model generalization through a Bayesian probabilistic framework. Its implementation in MATLAB as the `trainbr` function integrates the training and validation phases, providing an optimal solution for limited datasets common in groundwater contamination scenarios. This method reduces the risk of overfitting by automatically adjusting regularization parameters during training, effectively improving predictive accuracy and robustness.

Our approach simulates contaminant migration in a shallow groundwater aquifer and compares the performance of both algorithms while evaluating the influence of key hydrogeological parameters using MATLAB.

2 Methodology

2.1 Conceptual Model and Boundary Conditions

The 2D generic contamination model domain spans 150 m in length and 100 m in width, and is set to be located at a depth of 30 m (depth being just relevant to make clear that it represents a shallow aquifer). Two leakage sources are positioned along the left boundary, while an observation well is placed on the right side of the domain. The key input parameters for the simplified model are *permeability*, *porosity*, and *layer length*, while the output is the contaminant concentration.

Groundwater flows from left to right under single-phase flow conditions, with the contaminant treated as a conservative tracer (i.e., sorption and degradation processes are neglected). Pressure head boundaries are applied laterally, and no-flow conditions are enforced at the top and bottom boundaries. All strata are assumed to be homogeneous and isotropic. The contaminant is allowed to leak into the aquifer over a simulation period of 150 hours. *Figure 1* presents a detailed diagram of the proposed model.

2.2 Neural Network Structures and Training Algorithms

A FFNN with one hidden layer is employed to predict pollutant concentrations in shallow aquifers. Similar neural network structures have been applied in recent groundwater contamination studies by Sandra M. Guzman et. al. (2017) [2]. and Taherdangkoo et. al. (2020) [8].

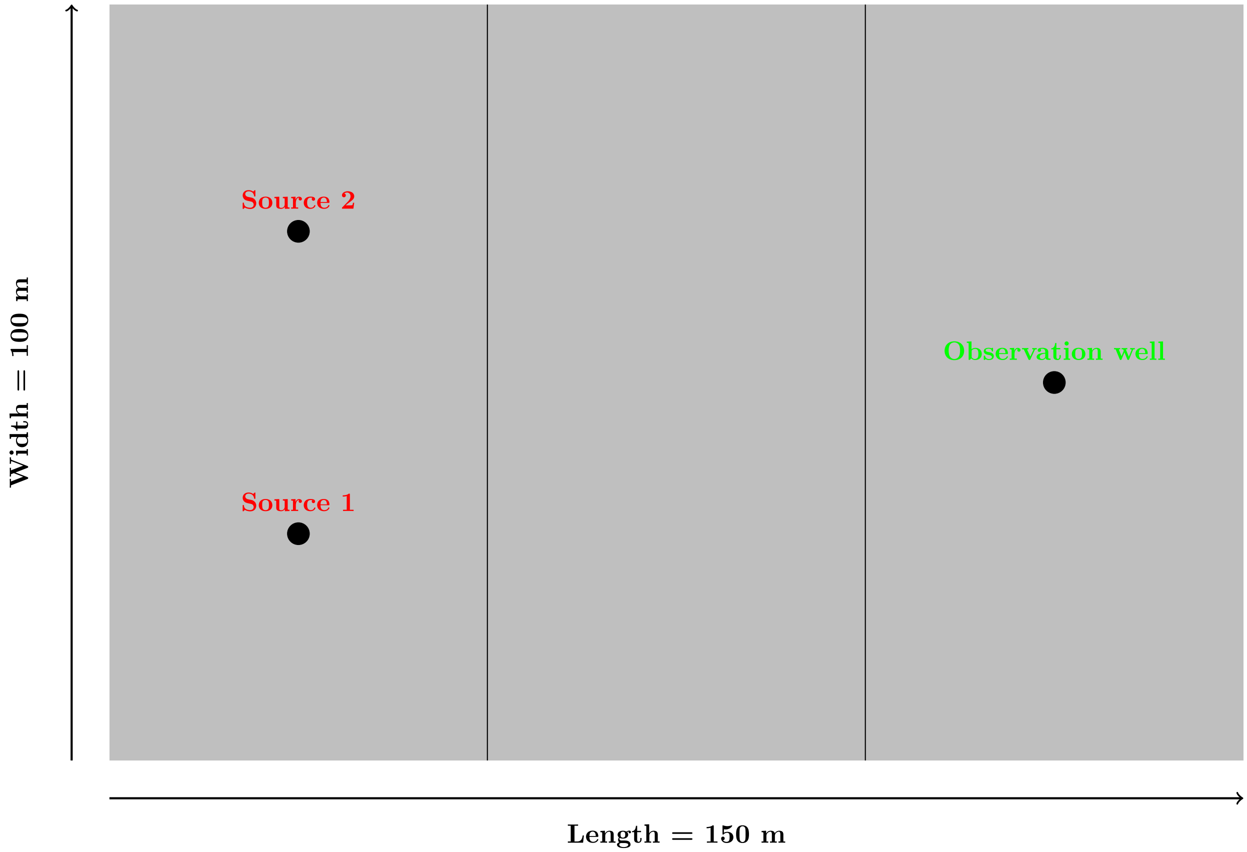

Figure 1: *2D generic contamination model to investigate flow and transport of contaminant plumes in a shallow aquifer. Modified in LaTeX from the WS 2023 Prof. Taherdangkoo slides.*

Data Normalization and Division

Input and target data are normalized to the range using the *mapminmax* function, since training algorithms typically perform better with data centered at zero [2, 8]. The dataset is split into training (70%), validation (15%), and testing (15%).

Network Architecture

The hidden layer uses a hyperbolic tangent sigmoid (*tansig*) [5] transfer function, chosen over the default *logsig* [6] due to the normalized data range of $[-1, 1]$. The output layer applies a linear transfer function to facilitate continuous-value prediction.

Training Algorithms

As explained earlier, the two algorithms used are the LM and the BR. LM algorithm is favored in the field, for fast convergence and efficiency, particularly in scenarios requiring rapid training and reliable predictions [8].

Regarding the FFNN-BR, which often integrates training and validation sets, enhancing generalization [1, 2, 8], we still separated the validation set. The reason was to ensure that both algorithms were evaluated under the same conditions. The validation partition is effectively inert for stopping in the BR case, but it still allows us to compute an even validation error metric for both algorithms.

Performance Evaluation

Model performance is assessed via performance metrics. The MSE, RMSE and R^2 are computed separately on the training, validation and test sets, per every number of neurons defining every model. The model yielding the lowest validation MSE is identified as the best model for each algorithm.

Parameter Importance Analysis

The Leave-One-Variable-Out (LOVO) method is executed in the code by systematically removing each input variable (*permeability*, *porosity*, and *layer length*) and retraining the best FFNN (previously selected depending on the validation MSE, as explained earlier) without that specific parameter. The method is simplified from J. D. Olden et. al. (2004) [7]. The performance degradation is quantified by computing the mean squared error (MSE) for training and testing datasets. The impact of each parameter is measured by averaging these MSE differences and normalizing them to derive a relative importance metric. Finally, a bar chart is generated to visualize the normalized contribution of each parameter, aiding in the interpretation of their influence on the model's predictions. Validation MSE is not included in this step, as it is primarily used for model selection rather than importance quantification. Including it could introduce bias, since the validation set is already involved in tuning the best FFNN, potentially leading to an overestimation or underestimation of parameter importance due to prior model adjustments.

3 Results and Discussion

Both FFNN-LM and FFNN-BR demonstrated strong predictive capabilities when trained on normalized data. The FFNN-LM achieved its best performance using 30 neurons (Training MSE: 8.77×10^3, Validation MSE: 4.42×10^4, Testing MSE: 1.04×10^5), while the FFNN-BR performed optimally with 13 neurons (Training MSE: 2.45×10^4, Validation MSE: 4.23×10^4, Testing MSE: 4.57×10^4). These results indicate that the FFNN-BR generally outperformed the FFNN-LM, particularly in terms of testing error and number of neurons.

To provide a more comprehensive assessment of predictive performance, additional evaluation metrics, including Root Mean Squared Error (RMSE) and the coefficient of determination (R^2), were computed.

Figure 2 presents the performance comparison between the FFNN-LM and FFNN-BR algorithms in terms of MSE as a function of the number of neurons. The results indicate that while LM achieves lower training error, the BR network exhibits superior generalization capabilities. In this graph, FFNN-LM obtains the highest peaks of MSE, especially for the testing dataset.

Figure 3 and *Figure 4* illustrate similar trends for RMSE and R^2 metrics. The RMSE comparison reveals that the BR network maintains lower error values across all tested neuron configurations, better than *Figure 2*. Likewise, *Figure 4* confirms that the BR network achieves higher R^2 values, signifying a stronger correlation between predicted and actual pollutant concentrations, also making the trends clearer than *Figure 2*. These graphs also support the selection criteria for the optimal model based on validation error. While alternative FFNN configurations, particularly LM with a lower number of neurons, could have been considered, our final choice remains justified, especially when evaluating the validation functions.

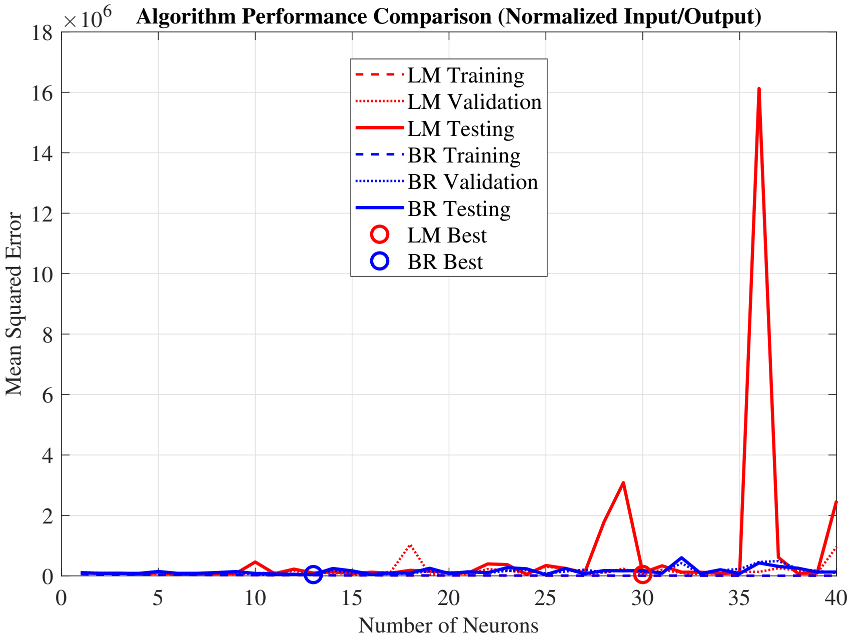

Figure 2: *Algorithm Performance Comparison: MSE vs. number of neurons for the FFNN-LM and FFNN-BR. Circular markers indicate the best-performing networks for each algorithm, pointing on the validation dataset line.*

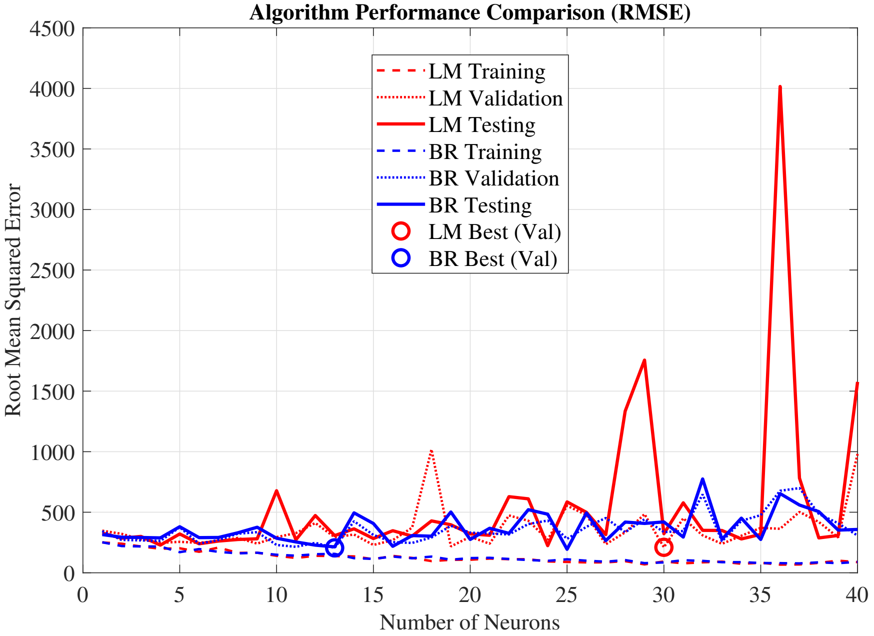

Figure 3: *Algorithm Performance Comparison: RMSE vs. number of neurons for the FFNN-LM and FFNN-BR. Circular markers indicate the best-performing networks for each algorithm, pointing on the validation dataset line.*

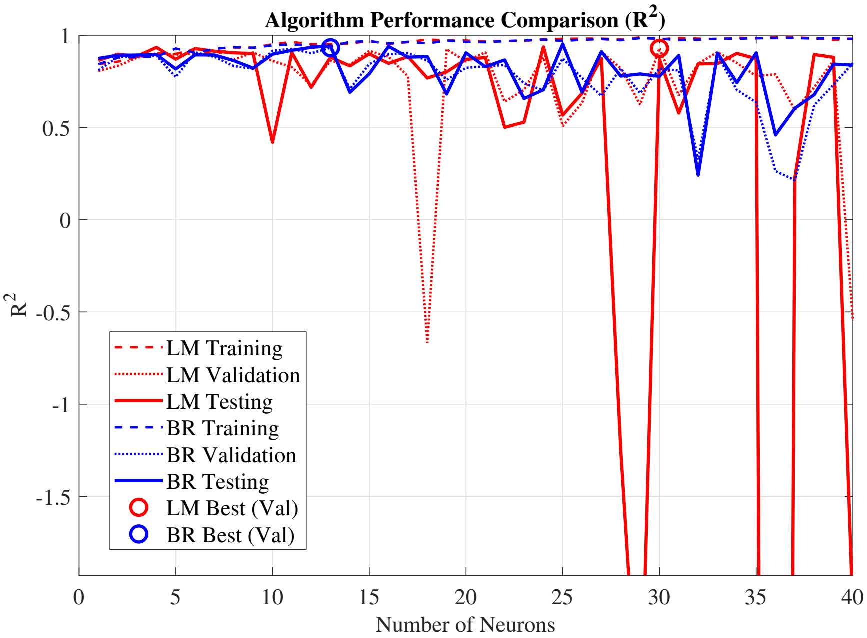

Figure 4: *Algorithm Performance Comparison: R^2 vs. number of neurons for the FFNN-LM and FFNN-BR. Circular markers indicate the best-performing networks for each algorithm, pointing on the validation dataset line.*

The regression analyses for the FFNN-LM and FFNN-BR, depicted in *Figure 5* and *Figure 6*, respectively, provide further insights into model performance. For the FFNN-LM (*Figure 5*), scatter plots of true versus predicted values indicate a strong correlation along the 45-degree line, though larger discrepancies appear in the validation and testing datasets. Similarly, *Figure 6* shows that the BR network exhibits better agreement between predicted and actual pollutant concentrations, confirming its superior predictive accuracy. The FFNN-LM also performs worse regarding extreme values, this is best observed from the 2000 line onwards, in both axis, where 9 outliers are identified in the testing dataset (with its particular importance, since it doesn't train the FFNNs).

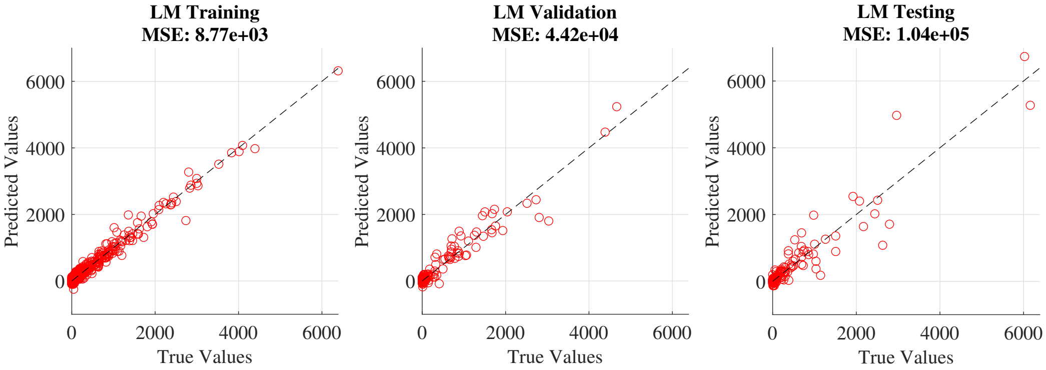

Figure 5: *LM Network Regression Analysis: Scatter plots for (a) training, (b) validation, and (c) testing datasets with corresponding MSE values.*

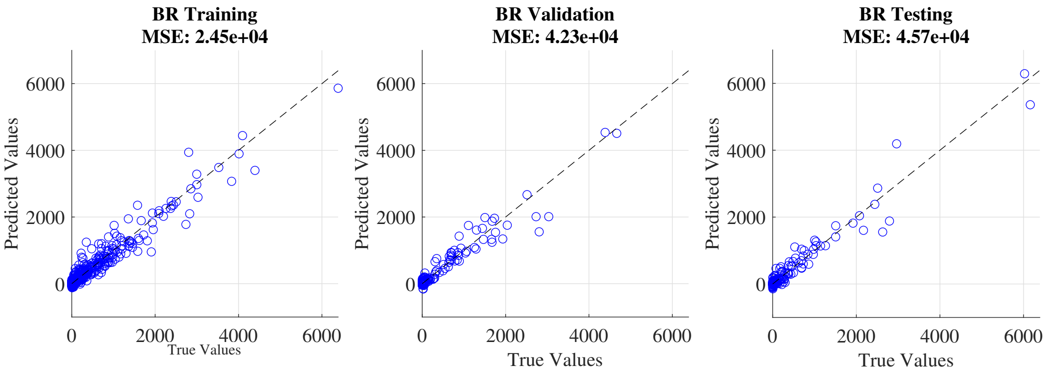

Figure 6: *BR Network Regression Analysis: Scatter plots for (a) training, (b) validation, and (c) testing datasets with corresponding MSE values.*

To assess parameter importance, a sensitivity analysis was conducted. The bar chart in *Figure 7* reveals that *porosity* is the most influential parameter (normalized importance: 0.543), followed by *layer length* (0.281) and *permeability* (0.176). These findings underscore the dominant role of *porosity* in determining pollutant dispersion, while *permeability* and *layer length* remain significant secondary contributors.

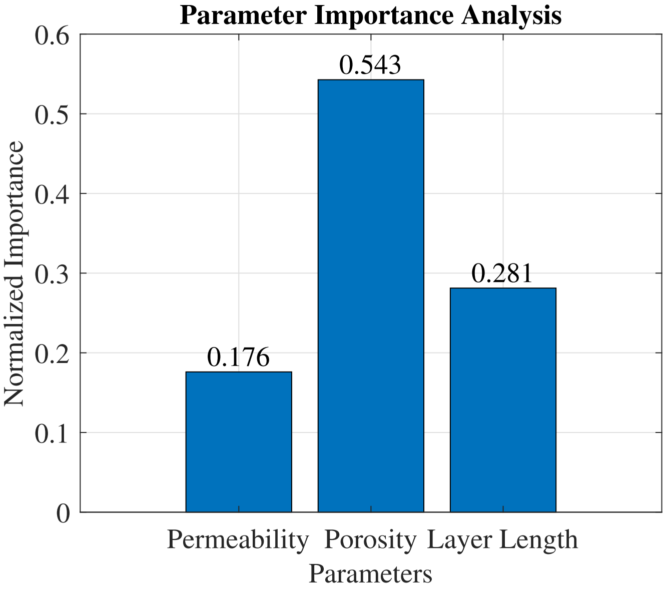

Figure 7: *Parameter Importance Analysis: Normalized importance of input parameters obtained by evaluating the impact of each parameter's exclusion on network performance.*

4 Conclusion

This study has demonstrated the application of data-driven neural network approaches for the prediction of pollution concentrations in a shallow groundwater aquifer. A two-dimensional contamination model was employed to simulate pollutant transport under realistic boundary conditions. Neural networks with varying numbers of hidden neurons were trained using both Levenberg-Marquardt and Bayesian Regularization algorithms.

Both LM and BR networks provided accurate predictions, but the results clearly indicate that the BR algorithm is the superior choice for this application in all ambits (less number of neurons, and less MSE, RMSE and higher R^2 across testing datasets). The FFNN-LM performance was optimized by monitoring validation MSE, while the BR network benefited from an automatic regularization mechanism, enhancing its generalization ability.

The parameter importance analysis confirmed that *porosity* is the most significant predictor of pollution spread, while *permeability* and *layer length* also play important roles. These findings provide valuable insights into groundwater contamination modelling and suggest that BR-based feedforward neural networks are well-suited for predictive environmental analysis.

References

[1] Foresee, F. D., & Hagan, M. T. (1997). Gauss–Newton approximation to Bayesian learning. In Proceedings of International Conference on Neural Networks (ICNN'97) (Vol. 3, pp. 1930–1935). IEEE. https://doi.org/10.1109/ICNN.1997.614194

[2] Guzman, Sandra M.; Paz, Joel O.; Tagert, Mary Love M. (2017): The Use of NARX Neural Networks to Forecast Daily Groundwater Levels. In Water Resour Manage 31 (5), pp. 1591–1603. DOI: 10.1007/s11269-017-1598-5.

[3] Hagan, M. T., & Menhaj, M. B. (1994). Training feedforward networks with the Marquardt algorithm. IEEE Transactions on Neural Networks, 5(6), 989–993. https://doi.org/10.1109/72.329697

[4] MathWorks. (n.d.). Choose a multilayer neural network training function. MATLAB Documentation. Retrieved March 7, 2025, from https://www.mathworks.com/help/deeplearning/ug/choose-a-multilayer-neural-network-training-function.html

[5] MathWorks (n.d.). tansig: Hyperbolic tangent sigmoid transfer function. Retrieved from https://de.mathworks.com/help/deeplearning/ref/tansig.html.

[6] MathWorks (n.d.). logsig: Log-sigmoid transfer function. Retrieved from https://de.mathworks.com/help/deeplearning/ref/logsig.html.

[7] Olden, J. D., Joy, M. K., & Death, R. G. (2004). An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. Ecological Modelling, 178(3–4), 389–397.

[8] Taherdangkoo, R., Tatomir, A., Taherdangkoo, M., Qiu, P., & Sauter, M. (2020). Nonlinear autoregressive neural networks to predict hydraulic fracturing fluid leakage into shallow groundwater. Water, 12(3), 841. https://doi.org/10.3390/w12030841