INTERNATIONAL JOURNAL OF INTEGRATED ENGINEERING
ISSN: 2229-838X e-ISSN: 2600-7916

IJIE

Vol. 17 No. 9 (2025) 337-352
https://publisher.uthm.edu.my/ojs/index.php/ijie

ANN Based Approach for Simultaneous Detection of Groundwater Pollution Origin and Release Characteristics

Himanshu Arora[1], Jyoti Chaubey[1]*, Shreyanshu Saha[1]

[1] *Department of Civil Engineering, MNIT Jaipur, Jaipur, 302017 INDIA*

*Corresponding Author: jyoti1810@gmail.com
DOI: https://doi.org/10.30880/ijie.2025.17.09.023

Article Info

Received: 4 May 2025
Accepted: 13 November 2025
Available online: 31 December 2025

Keywords

ANN, groundwater pollution, source identification, concentration breakthrough curves

Abstract

Present study aims at simultaneously identifying the three essential features of groundwater pollution source identification (GPSI) problem viz. the pollutant source location i.e. the distance between origin and observation point (x), duration of the pollutant release (T_0) and its concentration strength i.e. the pollutant release flux history (C_0). Four Artificial Neural Network (ANN) models were developed, the three models, viz. ANN1, ANN2, ANN3 identified the three source parameters individually; while the fourth model ANN4 aimed at identifying the source parameters x, C_0 and T_0 simultaneously. The model input comprised multiple sets of concentration breakthrough curves (BTCs) at 50 discrete 'x' for 10 different T_0 and 10 varying C_0 values. Employing the governing partial differential equation of pollutant transport in groundwater, 383130 patterns of BTCs were generated for training and testing of ANN models. Model performance was evaluated using standard statistical measures to recognize the optimal ANN architecture. A network with architecture 14-15-3 was found to be optimum for ANN4 and resulted in a average absolute relative error (AARE) of 9.17, 19.96, 16.75 for identifying x, C_0 and T_0 respectively. Single output ANN models performed better than the multiple output ANN model. While comparing for individual source parameters as target variables, both the models (single and multiple outputs) could efficiently identify the source location. Whereas, the release concentration and duration of pollutant release estimation by single output model performed better than the multiple output model. The proposed ANN based solution of the GPSI problem illustrated here, has a practical application in formulating strategies for regulating and penalizing the agencies accountable for the groundwater pollution.

1. Introduction

Increase in worldwide population has led to significant increase in the water consumption as well as wastewater generation in municipal, agricultural and industrial sectors. Unwise disposal of wastewater to the surface/groundwater sources lead to their contamination, and this issue has aggravated in the past few years due to compounding anthropogenic stress. As compared to surface water sources, groundwater is more resistant to contamination and pollution; however, if contaminated, its remediation is costly and is time consuming. Therefore, groundwater pollution source identification plays a pivotal role in conducting risk assessment studies, designing the groundwater remediation measures and developing groundwater management strategies.

Groundwater pollution source identification (GPSI) problem is defined as an inverse problem in the context of groundwater solute transport modeling. The source parameters responsible for groundwater pollution are characterized as the pollutant source location (x), its release duration (T_0) and released concentration flux history (C_0). The solution methodologies vary from complex physically based mathematical solutions to data driven black box models. Some of the widely adopted techniques of the GPSI problem solution involve deterministic direct methods such as maximum likelihood estimation [1]; Tikhonov regularization [2]; quasi-reversibility: reversed time solution of the governing equation [3]; minimizing the correlation coefficients [4]; marching jury backward beam equation [5]. Stochastic direct methods are random walk particle method [6]; minimum relative entropy inverse method [7]; geostatistical and Bayesian principles [8]–[10].

Indirect methods are based on optimization wherein a simulator and an optimizer are used together to solve GPSI problem [11]–[17]. Simulator employs either analytical or numerical solutions for the governing equations of solute transport in groundwater. Approximate simulators, also known as surrogate models, are used to reduce the computational effort in solving GPSI problem as compared to using a simulation model. As in a simulation based GPSI problem, simulation model is called number of times by an optimizer during the solution process [16], [18], [19]–[25].

Artificial Neural Network (ANN) is a widely used non-linear, data-driven machine-learning technique that captures the relationship between the input and output datasets in complex physical processes. ANN is also extensively used for solving the inverse GPSI problem. Singh & Datta [26] estimated the pollution source location and flux simultaneously using ANN. Bora & Bhattacharjya [27] utilized a hybrid approach, linking ANN with an optimizer for solving GPSI problem. In this method, ANN was employed to simulate flow and transport processes within the aquifer. Zhao *et al.* [18] used ANN, as a surrogate model linked with an optimizer to identify the source parameters. Leichombam Bhattacharjya [16] used ANN- modified GA based model to find the source location and source flux and release history. Secci *et al.* [28] and Chaubey & Srivastava [29] used ANN for solving the GPSI problem.

Majority of the mentioned GPSI studies evidently state that the location of the source fluxes has to be known (or to be assumed) in advance. This assumption of a-priori knowledge of source locations does not always hold good for the real field scenario. Some of the reported studies have identified the pollutant source locations but either the source location alone has been identified [30]–[32]; or location and source flux have been simultaneously identified [4], [33]. Concentration alone has been found by Leichombam & Bhattacharjya [16].

Present study deals with the development of GPSI models based on ANN. Pollution source parameters identified are: pollution source location/origin (distance between the source and the monitoring well bearing the existing pollutant concentration data), source release characteristics i.e. the duration of concentration release and its release concentration flux. Four ANN models are developed, three models viz. ANN1, ANN2 and ANN3 identify each source parameter individually while the fourth one 'ANN4' identifies the pollution source parameters simultaneously for given pollutant concentration data at advancing times at a monitoring well. The concentration data was available for a period of 10 years from the time of pollutant release at a temporal resolution of 1 day. Simplified ANN model architecture is attained through discretizing the concentration breakthrough curves in a specific manner and presenting it as input to the model. The database for training and testing of ANN models, comprising breakthrough curve as input and source parameters (mentioned previously) as output, are generated by employing a simulation model based upon the analytical solution of the groundwater solute transport equation [32], [34].

Present study focuses on the development of different ANN models aiming to identify unknown pollution source parameters (i.e. source location; duration of concentration release and released concentration flux): (i) individually (ii) simultaneously (all three at a time). Older studies have incorporated use of certain modelling and optimization techniques to find these parameters individually while only a few of them have worked on their simultaneous identification. This study tends to focus more on exploring simultaneous identification of the parameters which has not been much explored in the previous reported studies.

2. Methodology

The methodology involves the description of model development, and subsequently the illustration of the proposed models. The model development comprises the details of ANN model architecture that employs the simulation-model-generated-data to achieve the major target of this study.

2.1 ANN Model Architecture

In the present study, a feed-forward three-layered ANN model was used for solving GPSI problem. To simplify model architecture, the available time series data of pollutant concentration (BTC) from a monitoring well, was discretized in a specific way and fed into the ANN model as input along with other known source parameters. Three single output ANN model structures were proposed (viz. ANN1, ANN2 and ANN3), followed by one multiple output model (ANN4) for simultaneous identification of target variables. The outputs (target variables) for the

Int. Journal of Integrated Engineering Vol. 17 No. 9 (2025) p. 337-352

models are: for ANN1: pollutant release concentration at source (C_0); for ANN2: distance between the pollutant source and the monitoring well (x) and for ANN3: pollutant release duration by the source (T_0). For ANN4 all the three source parameters (C_0, x, T_0) were the outputs of the model. The ANN model architecture has 3 layers, viz. input, hidden and output. The data fed to the input layer of ANN undergoes forward processing as it traverses through the hidden layer, and ultimately resulting in the computation of output at the output layer. Models had single hidden layer with tan sigmoidal transfer function connecting the input and hidden layers, and a linear transfer function linking the hidden layer to the output layer. The weights are assigned initially and reiterated subsequently; and the number of neurons in hidden layer is varied, with an aim to achieve the best fitting ANN model structure. The inputs/output data to ANN model were wisely divided into training and testing parts and were used for model calibration and evaluation. Schematic representation of ANN model architecture for single output is given in Fig. 1.

Two different learning algorithms viz. Levenberg Marquardt backpropagation (LMBP) and Bayesian Regularization Back propagation (BRBP) were used in the single output ANN models in order to compare the running time, convergence rate [35]. Better performing learning-algorithm was further used for multi output ANN model. Bayesian regularization backpropagation learning algorithm is based on the objective function, mean square error (MSE), which is the mean of the square of the difference between the predicted and the actual observed values. In the BRBP algorithm, the objective function is as follows:

$$F(\omega) = \alpha E_\omega + \beta E_D \tag{1}$$

where: α and β = regularization coefficients; ω = neural network weights. $E_\omega = \sum_i \omega_i^2$ represents sum of squared network weights; E_D represents data error (difference between the predicted and observed values) term.

Bayesian regularization method is used to determine the optimal values for α and β. BRBP algorithm is based on incorporating prior distributions over the model parameters and sequentially updating them using techniques like variational inference or Markov Chain Monte Carlo (MCMC) methods.

The Levenberg–Marquardt training algorithm is aimed at minimizing a cost function, defined as the residual sum of squares (RSS) [35]. This cost function is used in nonlinear regression problems and the optimization of models. Furthermore, the model parameters are iteratively updated using a combination of descending gradient methods and Newton methods, given as:

$$w_{k+1} = w_k - \left(J_k^T J_k + \mu I\right)^{-1} J_k e_k \,;\; J_{ki} = \frac{\partial e_k}{\partial w_i} \tag{2}$$

where: μ = parameter governing the step size; e_k = error term representing the difference between the model predicted and observed value; k = subset of number of observations (n); J = Jacobian matrix. Very large values of λ refer to standard gradient descent, while very small values λ of refer to the Newton method.

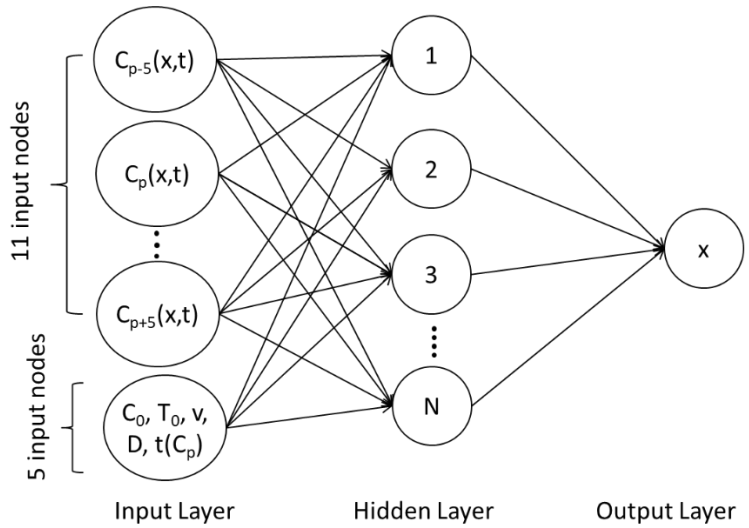

Fig. 1 *ANN model architecture for single output (ANN2)*

The performance of models is generally evaluated on the basis of certain statistical parameters which either show the fitting of modelled dataset with observed data or show residual error of the model. The statistical evaluation parameters incorporated in the study to evaluate the performance of model are coefficient of correlation (R), Nash Sutcliffe Efficiency (E) and Average Absolute Relative Error (AARE).

$$R = \frac{\sum\left(X_{obs} - \bar{X}_{obs}\right)\times\left(X_{\mathrm{mod}} - \bar{X}_{\mathrm{mod}}\right)}{\sqrt{\sum\left(X_{obs} - \bar{X}_{obs}\right)^2}\sqrt{\sum\left(X_{\mathrm{mod}} - \bar{X}_{\mathrm{mod}}\right)^2}} \tag{3}$$

$$E = 1 - \frac{\sum(X_{\mathrm{mod}} - X_{obs})^2}{\sum(X_{obs} - \bar{X}_{obs})^2} \tag{4}$$

$$AARE = \frac{1}{n}\sum\left|\frac{X_{\mathrm{mod}} - X_{obs}}{X_{obs}}\right| \times 100\% \tag{5}$$

where X_{obs} = observed data series, X_{mod} = model estimated data series, $\bar{X}_{obs}$ = average of observed data series, $\bar{X}_{\mathrm{mod}}$ = average estimated data series.

Coefficient of Correlation (R) measures the correlation or the linear relationship between datasets. Its range is between -1 to +1, the latter showing good linear dependency. The range of E varies from $-\infty$ and 1.0. For an ideal model where estimation error variance equal to zero, the resulting Nash–Sutcliffe Efficiency equals 1. AARE serves as a suitable loss function in regression analysis because of its ability to demonstrate the existence of an optimal model and the consistency of empirical risk reduction. The percentage error for too low forecasts cannot exceed 100%, whereas for too high forecasts, there is no upper limit to the percentage error.

The ANN model requires substantial dataset as input and output. Field data is required for this purpose so that the ANN could generate the results which imitate the real field scenarios. In absence of field data, the simulation model is employed to generate the required dataset. Various combinations of input parameters are given to simulation model (as input) to generate corresponding BTCs (as its output), which will provide the input to the ANN models. Details of the simulation model for the dataset generation are provided in the following section.

2.2 Simulation Model

The simulation model for generating the training and testing patterns for ANN model is based on the governing mass transport equation. This equation applies to a saturated, homogeneous and semi-infinite aquifer with uniform one-dimensional flow and conservative pollutant (Eq. 6). These assumptions of homogeneity, one dimension, uniformity etc. were made in order to reduce the computations in an otherwise complex process.

$$\frac{\partial C(x,t)}{\partial t} = D\frac{\partial^2 C(x,t)}{\partial x^2} - v\frac{\partial C(x,t)}{\partial x} \tag{6}$$

where, $C(x,t)$ = pollutant concentration (ML^{-3}) at distance x (L) from source at time t (T), D = dispersion coefficient (L^2T^{-1}) and v = groundwater flow velocity (LT^{-1}).

The initial condition, for obtaining the solution of governing differential equation (Eq. 6), is that the aquifer is uncontaminated in the beginning (i.e. no pollution in the beginning). The boundary conditions are derived assuming that the source released the pollutant at a constant flux for time period (T_0) and discontinued thereafter. Mathematically, the initial and boundary conditions are expressed as:

$$C(x,0) = 0\ ;\ C(0,t) = C_0 \text{ for } t \le T_0 \text{ and } 0 \text{ otherwise}\ ;\ C(\infty,t) = 0 \tag{7}$$

where, T_0 = duration during which pollutant was released (T) and C_0 = pollutant release concentration at source (ML^{-3}). The solution of Eq. 6, subjected to the initial and boundary conditions given in Eq. 7, is [34]

$$C(x,t) = \frac{C_0}{2}\left[erfc\left(\frac{x - vt}{\sqrt{4Dt}}\right) + e^{\frac{vx}{D}}erfc\left(\frac{x + vt}{\sqrt{4Dt}}\right)\right] \text{ for } t \le T_0 \text{ (Source still emitting pollutant)} \tag{8}$$

Int. Journal of Integrated Engineering Vol. 17 No. 9 (2025) p. 337-352

$$C(x,t)=\frac{C_0}{2}\left[erfc\left(\frac{x-vt}{\sqrt{4Dt}}\right)+e^{\frac{vx}{D}}erfc\left(\frac{x+vt}{\sqrt{4Dt}}\right)\right]-\frac{C_0}{2}\left[erfc\left(\frac{x-v(t-T_0)}{\sqrt{4D(t-T_0)}}\right)+e^{\frac{vx}{D}}erfc\left(\frac{x+v(t-T_0)}{\sqrt{4D(t-T_0)}}\right)\right]$$

for $t > T_0$ (When the source stops emitting pollutant) (9)

where,

$$erfc(x)=1-\left(\frac{2}{\pi^{0.5}}\right)\int_0^x e^{u^2}du \tag{10}$$

Eq. 8 and 9 were employed to generate breakthrough curves. A breakthrough curve is a simple curve between the concentrations of contaminant at a certain location with respect to time. Symbolically, it can be expressed as:

$$C(x,t)=f(x,t,D,v,C_0,T_0) \tag{11}$$

i.e., for every individual set of *x*, *D*, *v*, C_0, T_0 values; a distinct/unique breakthrough curve is generated. For every changed value of *x*, a new breakthrough curve is generated while for every change in the other parameters new patterns, each consisting of a certain number of breakthrough curves (dependent on variation of *x*), were generated.
The parameters *x*, *D*, *v*, C_0, T_0 were varied within a certain range (Table 1) and then used as an input in the solution of 1-D mass transport equation to generate a number of breakthrough curves.

Table 1 *Ranges for variation of parameters considered*

S. No	Parameters (unit)	Variation of parameters	Number of data points
1.	x(m)	$x_1, x_2 \ldots\ldots x_n$	n
2.	C_0 (mg/l)	$C_{01}, C_{02} \ldots\ldots C_{0m}$	m
3.	T_0 (days)	$T_{01}, T_{02} \ldots\ldots T_{0l}$	l
4.	v (m/day)	$V_1, V_2 \ldots\ldots V_p$	p
5.	D (m^2/day)	$D_1, D_2 \ldots\ldots D_q$	q
6.	t (days)	$t_1, t_2 \ldots\ldots t_r$	r

The number of breakthrough curve patterns generated based on the variation given in Table 1 is (m (C_0 values) × l (T_0 values) × p (*v* values) × q (*D* values)) where each pattern consists of n (*x* values) breakthrough curves. Each breakthrough curve comprises of about r (*t* values) discrete values from which certain $C(x,t)$ values were selected. Eleven discrete values of $C(x,t)$ from a BTC, which includes the peak value (C_p) along with five successor and predecessor values of the peak along with the time value corresponding to peak concentration ($t(C_p)$) were selected as input variables. However, in some cases the value of peak may not occur within the observation period (*t*) of the contaminant source or the discrete values of the curve being negligible. Hence these cases were removed from the input dataset. The other input variables comprised of various combinations of variables as shown in Table 2.

Table 2 *Description of input and output variables for various ANN models*

Model	No. of input variables	Input Variables	Output
ANN1	16	C (x, t), x, T_0, v, D, $t(C_p)$	C_0
ANN2	16	C (x, t), C_0, T_0, v, D, $t(C_p)$	x
ANN3	16	C (x, t), C_0, x, v, D, $t(C_p)$	T_0
ANN4	14	C(x, t), v, D, $t(C_p)$	x, C_0, T_0

A schematic representation of the methodology adopted for proposed models is shown in Fig. 2.

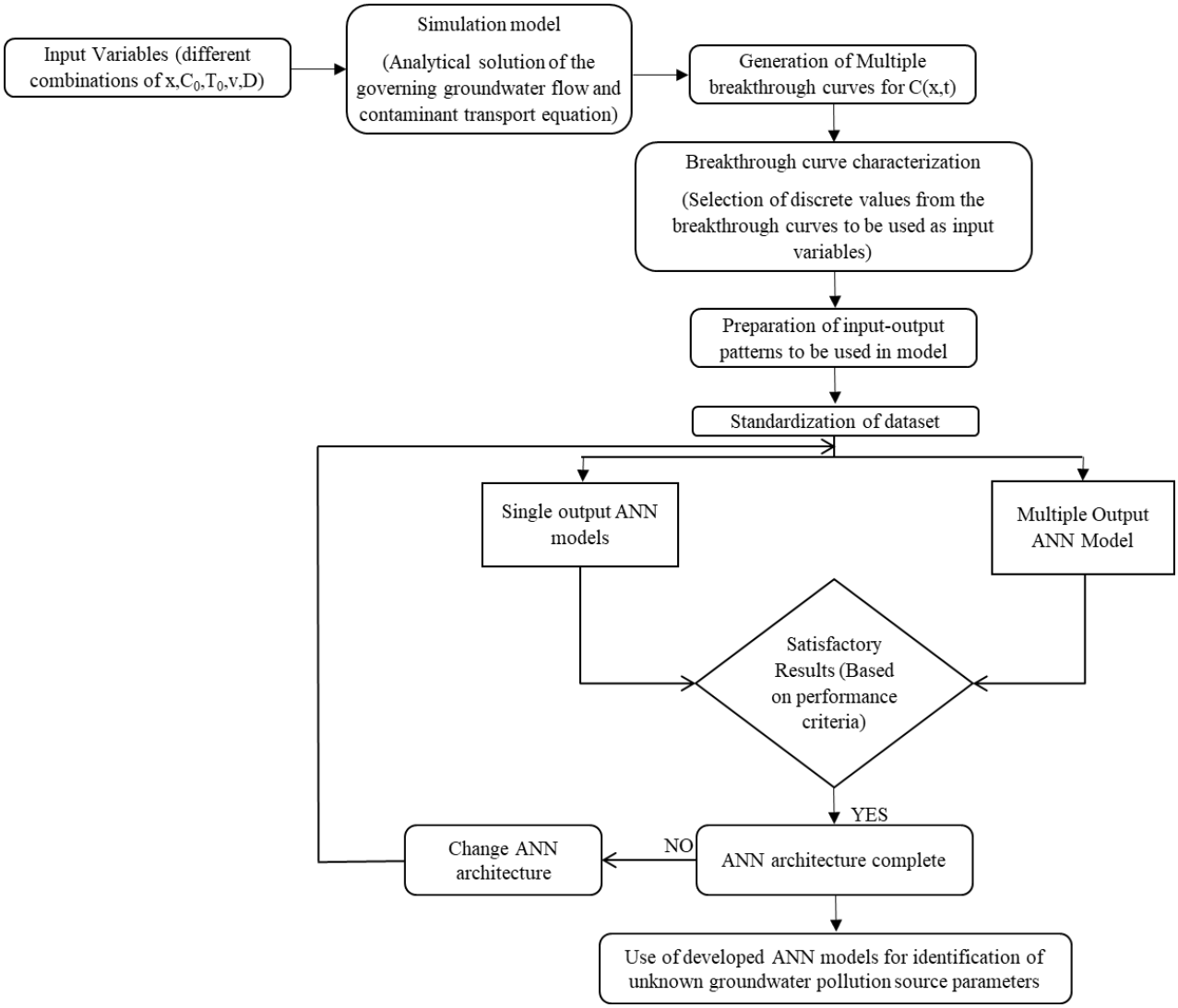

Fig. 2 *Schematic representation of the adopted methodology for the proposed models*

2.3 Model Illustration

Proposed ANN models are illustrated with the hypothetical problem worked upon by Kumar *et al.* [32]. A semi-infinite, uniform, one-dimensional aquifer domain is considered. It was assumed that the pollution source is releasing a conservative pollutant at constant concentration flux (C_0) at the left boundary, for a certain constant duration (T_0). The shape of a BTC ($C(x,t)$) depends on the parameters x, C_0, T_0, v and D. A unique BTC is generated for a unique combination the above-mentioned parameters. In this study multiple BTCs were generated by varying x, C_0, T_0, v and D; in a predefined range and variation pattern (as shown in Table 3); for a time span of 3650 days (10 years). The certain discrete information from the generated BTCs were used as input for ANN based model in identification for contaminant source location.

Table 3 *Variation of the different parameters for generation of input dataset*

S. No.	Parameters (units)	Variation of parameters	Number of data points
1.	x (m)	10, 20, 30, …., 500	50
2.	C_0 (mg/l)	100, 200, 300, …., 1000	10
3.	T_0 (days)	100, 200, 300, …., 1000	10
4.	v (m/day)	0.05, 0.06, 0.07, …., 0.15	11
5.	D (m^2/day)	0.05, 0.06, 0.07, …., 0.15	11
6.	t (days)	1, 2, 3, …., 3650	3650

Int. Journal of Integrated Engineering Vol. 17 No. 9 (2025) p. 337-352

A total of 12,100 (10 (C_0 values) × 10 (T_0 values) × 11 (v values) × 11 (D values)) BTCs patterns were generated resulting in a total of 605000 breakthrough curves i.e., a single pattern consisted of 50 different BTCs (one for each value of x). After filtering and removing the curves with negligible values and ones whose peak wasn't achieved in the observation period, the number of BTCs reduced to 383130. The training algorithm, Levenberg Marquardt backpropagation (LMBP), used about 229908 curves for training (60% of total data), 76636 curves for validation (20% of total data) and rest for testing (20% of total data) the single output ANN models. The other training algorithm Bayesian Regularization backpropagation (BRBP) used 306544 curves for training (80% of total data) and the rest 20% of dataset for testing the model. The former algorithm although being faster has provided less accurate results compared to latter so its use was limited to single output models only. A Sample of BTC patterns generated at different locations (for D = 0.1 m²/day, v = 0.1 m/day, T_0 = 300 days) is shown in Fig. 3.

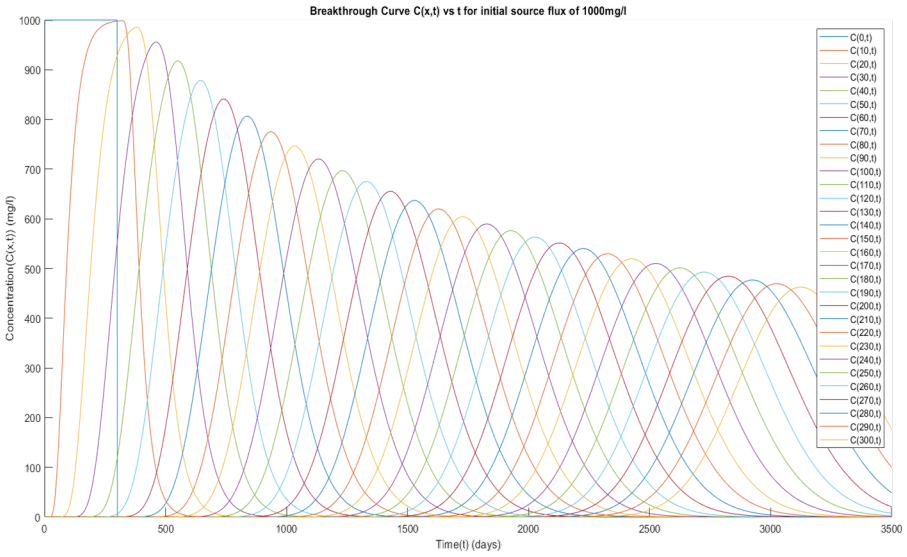

Fig. 3 *A sample breakthrough curve (C(x,t)) for an initial source flux (C_0) of 1000 mg/l*

The single output ANN models (ANN1, ANN2 and ANN3) are 16-N-1 models and three outputs ANN model (ANN4) was 14-N-3 model. Optimal number of hidden neurons N was determined based on the performance parameters. The results of all the architectures of ANN models are presented in a tabulated manner with the three evaluation parameters: Coefficient of correlation (R), Nash-Sutcliffe Efficiency (E) and Absolute Average Relative error (AARE). The best results are graphically represented for both training and testing datasets. The parameters used in models for predictands are illustrated in the table defined earlier (Table 3).

3. Results and Discussion

The results for the single output ANN models (ANN1, ANN2 & ANN3) and triple-output ANN model (ANN4) are included in the following sub-sections.

3.1 ANN1 Model Structure: C_0 = f (C (x_i, t_i), x, v, D, T_0, $t(C_p)$)

The model provided good results for both training and testing for the datasets, with minimal variation in model estimated data, with some outliers at higher values of C_0. Of the two training algorithms used, both were effective with similar results.

From Tables 4 and 5, it can be seen that the change in E and R with respect to the increase in number of hidden neurons was not significant. However, the AARE decreased as the number of hidden neurons increased from 5 to 23, but showed an increase when the neurons were further increased from 23 to 25. This was the case for both training and testing datasets and for both training algorithms. Therefore, the optimal number of neurons in hidden layer is assumed to be 23. The optimal ANN1 model architecture is 16-23-1 for both the training algorithms. Regression plots of the target release concentration (observed dataset) and the model estimated release concentration (model output) are shown in Fig. 4(a) and (b) (using LMBP as training algorithm). Fig. 4(c) and (d) show the regression plots for C_0 during training and testing, using BRBP as training algorithm. E and R values show that the observed and modelled values of C_0 remained in good agreement and varied minimally with the change in model architecture. This could be because of the usage of unperturbed data set which was presented to the models.

Table 4 *ANN1 model performance for C0 with different ANN architectures & LMBP training algorithm*

LMBP	Iteration	Training			Testing		
ANN1		E	R	AARE (%)	E	R	AARE (%)
16-5-1	496	0.991	0.996	5.31	0.991	0.996	5.30
16-7-1	1000	0.997	0.999	3.02	0.997	0.999	3.02
16-9-1	560	0.997	0.999	2.68	0.998	0.999	2.68
16-11-1	1000	0.999	0.999	1.98	0.999	0.999	1.98
16-13-1	929	0.999	0.999	1.44	0.999	0.999	1.44
16-15-1	999	0.999	0.999	1.51	0.999	0.999	1.50
16-17-1	999	0.999	0.999	0.97	0.999	0.999	0.97
16-19-1	1000	0.999	0.999	0.65	0.999	0.999	0.65
16-21-1	1000	0.999	0.999	0.66	0.999	0.999	0.66
16-23-1	1000	0.999	0.999	0.56	0.999	0.999	0.55
16-25-1	1000	0.999	0.999	0.74	0.999	0.999	0.74

Table 5 *ANN1 model performance for C0 with different ANN architectures & BRBP training algorithm*

BRBP	Iteration	Training			Testing		
ANN1		E	R	AARE (%)	E	R	AARE (%)
16-5-1	1000	0.991	0.996	5.42	0.991	0.996	5.40
16-7-1	1000	0.997	0.999	2.93	0.997	0.999	2.93
16-9-1	1000	0.998	0.999	2.24	0.998	0.999	2.24
16-11-1	1000	0.999	0.999	2.16	0.999	0.999	2.16
16-13-1	1000	0.999	0.999	1.15	0.999	0.999	1.14
16-15-1	1000	0.999	0.999	1.40	0.999	0.999	1.40
16-17-1	1000	0.999	0.999	1.10	0.999	0.999	1.11
16-19-1	1000	0.999	0.999	0.65	0.999	0.999	0.65
16-21-1	1000	0.999	1.000	0.64	0.999	1.000	0.64
16-23-1	1000	0.999	1.000	0.56	0.999	1.000	0.56
16-25-1	1000	0.999	0.999	0.63	0.999	0.999	0.63

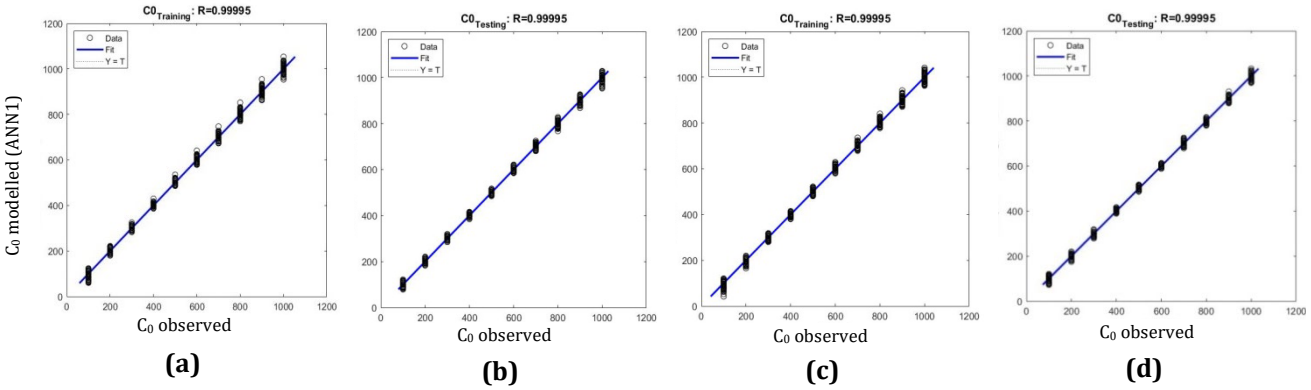

Fig. 4 *Regression plots of parameter C_0, for ANN based model (a) in training; and (b) testing by LMBP algorithm; and (c) in training; and (d) testing by BRBP algorithm*

3.2 ANN2 Model Structure: $x = f(C(x_i, t_i), v, D, C_0, T_0, t(C_p))$

The model provided good results for both training and testing for the datasets, with the variation of modelled dataset being minimal. The two different training algorithms used, both were effective however the BRBP algorithm showed better convergence with lesser values of AARE.

From Tables 6 and 7, it can be seen that AARE decreased as the number of hidden neurons increased from 5 to 21 but when the number of neurons increased from 21 to 23, AARE increased. This was the case for both training and testing datasets and for both training algorithms. Therefore, the optimal number of neurons in hidden

Int. Journal of Integrated Engineering Vol. 17 No. 9 (2025) p. 337-352

layer is assumed to be 21. The optimal ANN2 model architecture is 16-21-1 for both the training algorithms. Regression plots of the target source location (observed dataset) and the model estimated source location (model output) are shown in Fig. 5(a) and (b) (applying LMBP was used as training algorithm). Fig. 5(c) and (d) shows the regression plots for '*x*' during training and testing, using BRBP as training algorithm.

Table 6 *ANN2 model performance for x with different ANN architectures & LMBP training algorithm*

LMBP	Iteration	Training			Testing		
ANN2		E	R	AARE (%)	E	R	AARE (%)
16-5-1	1000	0.999	0.999	2.48	0.999	0.999	2.39
16-7-1	285	0.999	0.999	2.54	0.999	0.999	2.41
16-9-1	465	0.999	1.000	1.50	0.999	1.000	1.49
16-11-1	1000	1.000	1.000	0.78	1.000	1.000	0.80
16-13-1	1000	1.000	1.000	0.84	1.000	1.000	0.85
16-15-1	570	1.000	1.000	0.62	1.000	1.000	0.60
16-17-1	1000	1.000	1.000	0.89	1.000	1.000	0.91
16-19-1	843	1.000	1.000	0.44	1.000	1.000	0.42
16-21-1	1000	1.000	1.000	0.38	1.000	1.000	0.38
16-23-1	739	1.000	1.000	0.45	1.000	1.000	0.44
16-25-1	1000	1.000	1.000	0.37	1.000	1.000	0.37

Table 7 *ANN2 model performance for x with different ANN architectures & BRBP training algorithm*

BRBP	Iteration	Training			Testing		
ANN2		E	R	AARE (%)	E	R	AARE (%)
16-5-1	1000	1.000	1.000	0.61	1.000	1.000	0.61
16-7-1	1000	1.000	1.000	0.91	1.000	1.000	0.86
16-9-1	1000	0.999	1.000	1.31	0.999	1.000	1.28
16-11-1	1000	1.000	1.000	0.70	1.000	1.000	0.69
16-13-1	1000	1.000	1.000	0.48	1.000	1.000	0.51
16-15-1	1000	1.000	1.000	0.61	1.000	1.000	0.61
16-17-1	1000	1.000	1.000	0.41	1.000	1.000	0.41
16-19-1	1000	1.000	1.000	0.29	1.000	1.000	0.28
16-21-1	1000	1.000	1.000	0.26	1.000	1.000	0.26
16-23-1	1000	1.000	1.000	0.28	1.000	1.000	0.27
16-25-1	1000	1.000	1.000	0.42	1.000	1.000	0.42

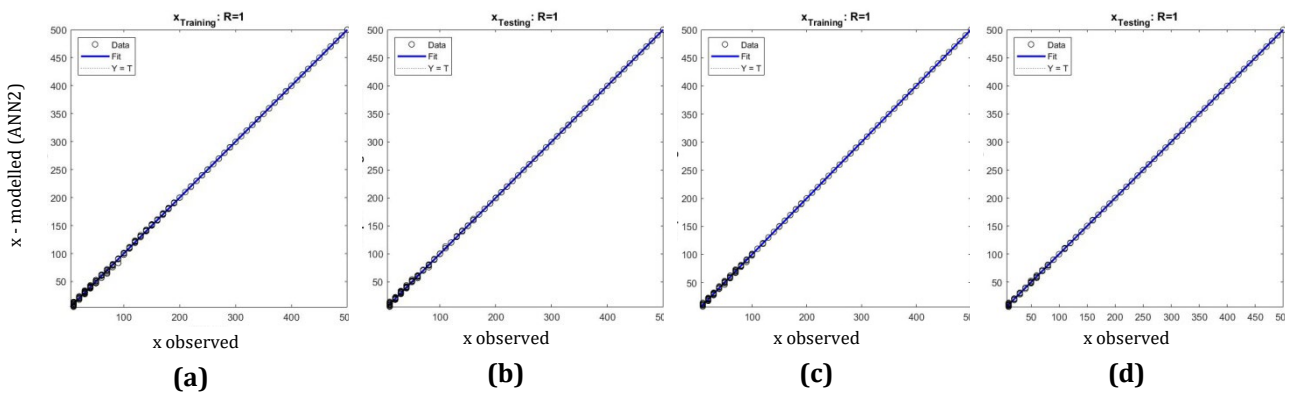

Fig. 5 *Regression plots of parameter x, for ANN based model (a) in training; and (b) testing by LMBP algorithm; and (c) in training; and (d) testing by BRBP algorithm*

3.3 ANN3 Model Structure: T_0 = f (C (x_i, t_i), x, v, D, C_0, $t(C_p)$)

The model provided good results for both training and testing for the datasets, with minimal variation in modelled data, with some outliers at higher values of T_0. Of the two training algorithms used, both were effective with similar results.

From Tables 8 and 9, it can be seen that AARE decreased as the number of hidden neurons increased from 5 to 23 but when the number of neurons increased from 23 to 25, AARE increased. This was the case for both training and testing datasets and for both training algorithms. Therefore, the optimal number of neurons in hidden layer is assumed to be 23. The optimal ANN3 model architecture is 16-23-1 for both the training algorithms. Regression plots of the target time of pollutant release (observed dataset) and the model estimated time of pollutant release (model output) are shown in Fig. 6(a) and (b) (using LMBP as training algorithm) and Fig. 6(c) and (d) (using BRBP as training algorithm). Model worked fine for lower values of time of release of pollutant from the source. While for higher values of T_0, deviation of the estimated values was found more that the observed data.

Table 8 *ANN3 model performance for T0 with different ANN architectures & LMBP training algorithm*

LMBP	Iteration	Training			Testing		
Model (T_0)		E	R	AARE (%)	E	R	AARE (%)
16-5-1	1000	0.991	0.996	5.23	0.991	0.996	5.24
16-7-1	681	0.994	0.997	3.90	0.994	0.997	3.89
16-9-1	1000	0.996	0.998	2.57	0.996	0.998	2.57
16-11-1	1000	0.997	0.999	1.61	0.997	0.999	1.61
16-13-1	1000	0.999	0.999	0.78	0.998	0.999	0.77
16-15-1	1000	0.999	0.999	0.82	0.999	0.999	0.83
16-17-1	1000	0.998	0.999	0.58	0.999	0.999	0.58
16-19-1	1000	0.999	0.999	0.54	0.999	0.999	0.31
16-21-1	1000	0.999	0.999	0.59	0.999	0.999	0.57
16-23-1	1000	0.999	0.999	0.51	0.999	0.999	0.51
16-25-1	1000	0.999	0.999	0.61	0.999	0.999	0.61

Table 9 *ANN3 model performance for T0 with different ANN architectures & BRBP training algorithm*

BRBP	Iteration	Training			Testing		
Model (T_0)		E	R	AARE (%)	E	R	AARE (%)
16-5-1	1000	0.966	0.983	11.42	0.966	0.983	11.40
16-7-1	1000	0.997	0.998	1.81	0.997	0.999	1.77
16-9-1	1000	0.997	0.998	1.86	0.997	0.998	1.89
16-11-1	1000	0.998	0.999	1.19	0.998	0.999	1.19
16-13-1	1000	0.999	0.999	0.78	0.998	0.999	0.77
16-15-1	729	0.999	0.999	0.82	0.999	0.999	0.83
16-17-1	1000	0.998	0.999	0.58	0.999	0.999	0.58
16-19-1	1000	0.999	0.999	0.54	0.999	0.999	0.31
16-21-1	1000	0.999	0.999	0.59	0.999	0.999	0.57
16-23-1	593	0.999	0.999	0.51	0.999	0.999	0.51
16-25-1	1000	0.999	0.999	0.61	0.999	0.999	0.61

Int. Journal of Integrated Engineering Vol. 17 No. 9 (2025) p. 337-352

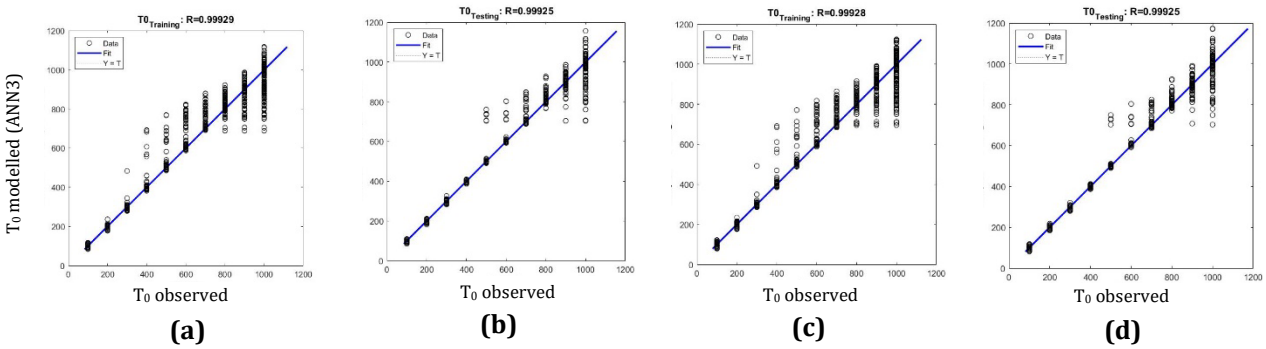

Fig. 6 *Regression plots of parameter T_0, for ANN based model (a) in training; and (b) testing for LMBP algorithm; and (a) in training; and (b) testing for BRBP algorithm*

Two different training algorithms were used in the single output ANN models in order to compare the running time, convergence rate and its applicability to the present study in order to choose better algorithm for further use. BRBP though having a longer iteration running time showed a faster convergence rate and was more suitable in solving this problem than LMBP. The pollution source parameter x was predicted well by BRBP while both training algorithms performance remained same for pollution source parameters C_0 and T_0. Thus, for three output model ANN4, BRBP was used as training algorithm.

Results of the three models ANN1, ANN2 and ANN3 varied minimally with change in model architecture, in terms of E and R values. But AARE values changed significantly with respect to the change in ANN architecture. Therefore, the optimal model architecture was adopted with respect to the change in AARE values. Models showed nearly perfect agreement between observed and modelled values of T_0 during training and testing. This could be because of the usage of unperturbed data set which was presented to the models.

3.4 ANN4 Model Structure: $(x, C_0, T_0) = f(C(x_i, t_i), V, D, t(C_p))$

The model showed good results for all the three parameters and was the best performing model of the all three output models. Of the three parameters x showed good results while C_0 and T_0 produced average results. The best architecture for the model was 14-15-3.

From Tables 10, 11 and 12, it can be seen that AARE decreased as the number of hidden neurons increased from 5 to 15 but when the number of neurons increased from 15 to 25, AARE increased. This was the case for both training and testing datasets and for both training algorithms. Therefore, the optimal number of neurons in hidden layer is assumed to be 15. The optimal ANN4 model architecture is 14-15-3. Regression plots of the target parameters (C_0, x, T_0) and the model estimated parameters are shown in Fig. 7. Results show that ANN4 model worked well for identifying the groundwater pollution source location (x) than compared to the identification of other two pollution source parameters.

Simultaneous identification of all the three unknown parameters was challenging since the degree of freedom of the model was compromised. The higher number of predictands increased the complexity and non-linearity of the problem, which the ANN model was able to regenerate marginally. The multiple-output-model estimated pollutant source location (x) was found to be in close agreement with the observed source location (Fig. 7 (c) and (d)). But the estimated C_0 and T_0 had shown a weak agreement along with a wide spread of data from the perfect-agreement trend-line, and hence it could be inferred that it did not match well with the respective observed values (Fig. 7(a), (b), (e) and (f)). The possible reason for this could be the neural network parameters becoming overly tailored to the training data, a condition known as overfitting and therefore the model overestimated the source release concentration and release duration.

Table 10 *ANN4 model performance for parameter C0 with different ANN architectures*

ANN4 for C_0	Iteration	Training			Testing		
		E	R	AARE (%)	E	R	AARE (%)
14-5-3	1000	0.810	0.900	25.21	0.811	0.900	25.20
14-7-3	1000	0.819	0.905	24.48	0.820	0.906	24.45
14-9-3	1000	0.829	0.910	23.82	0.830	0.910	23.75
14-11-3	1000	0.854	0.924	22.49	0.854	0.924	22.49
14-13-3	1000	0.860	0.927	21.60	0.860	0.927	21.57
14-15-3	1000	0.873	0.935	19.96	0.873	0.935	20.00
14-17-3	1000	0.864	0.929	20.24	0.864	0.929	20.25
14-19-3	1000	0.871	0.933	19.74	0.871	0.933	19.78
14-21-3	1000	0.871	0.933	18.96	0.871	0.933	18.93
14-23-3	1000	0.877	0.936	19.47	0.877	0.936	19.44
14-25-3	1000	0.866	0.930	19.23	0.866	0.930	19.24

Table 11 *ANN4 model performance for parameter x with different ANN architectures*

ANN4 for x	Iteration	Training			Testing		
		E	R	AARE (%)	E	R	AARE (%)
14-5-3	1000	0.977	0.989	21.25	0.977	0.989	20.77
14-7-3	1000	0.994	0.997	11.97	0.994	0.997	11.78
14-9-3	1000	0.993	0.997	12.07	0.993	0.997	11.92
14-11-3	1000	0.990	0.995	15.12	0.990	0.995	14.88
14-13-3	1000	0.988	0.994	15.30	0.988	0.994	15.54
14-15-3	1000	0.996	0.998	9.17	0.996	0.998	9.13
14-17-3	1000	0.994	0.997	11.15	0.994	0.997	11.06
14-19-3	1000	0.994	0.997	11.23	0.994	0.997	10.92
14-21-3	1000	0.993	0.996	11.73	0.993	0.996	11.47
14-23-3	1000	0.994	0.997	11.84	0.994	0.997	11.57
14-25-3	1000	0.993	0.996	13.01	0.993	0.996	12.76

Table 12 *ANN4 model performance for parameter T0 with different ANN architectures*

ANN4 for T_0	Iteration	Training			Testing		
		E	R	AARE (%)	E	R	AARE (%)
14-5-3	1000	0.837	0.915	28.21	0.837	0.915	28.20
14-7-3	1000	0.879	0.939	22.33	0.880	0.993	22.18
14-9-3	1000	0.880	0.938	22.33	0.882	0.939	22.23
14-11-3	1000	0.904	0.951	19.59	0.904	0.951	19.48
14-13-3	1000	0.914	0.956	19.11	0.914	0.957	19.02
14-15-3	1000	0.936	0.968	16.75	0.936	0.968	16.63
14-17-3	1000	0.911	0.955	19.15	0.911	0.955	19.17
14-19-3	1000	0.910	0.954	19.44	0.909	0.953	19.33
14-21-3	1000	0.900	0.948	19.68	0.900	0.949	19.54
14-23-3	1000	0.907	0.953	18.99	0.908	0.953	18.93
14-25-3	1000	0.896	0.947	20.35	0.897	0.947	20.29

Int. Journal of Integrated Engineering Vol. 17 No. 9 (2025) p. 337-352

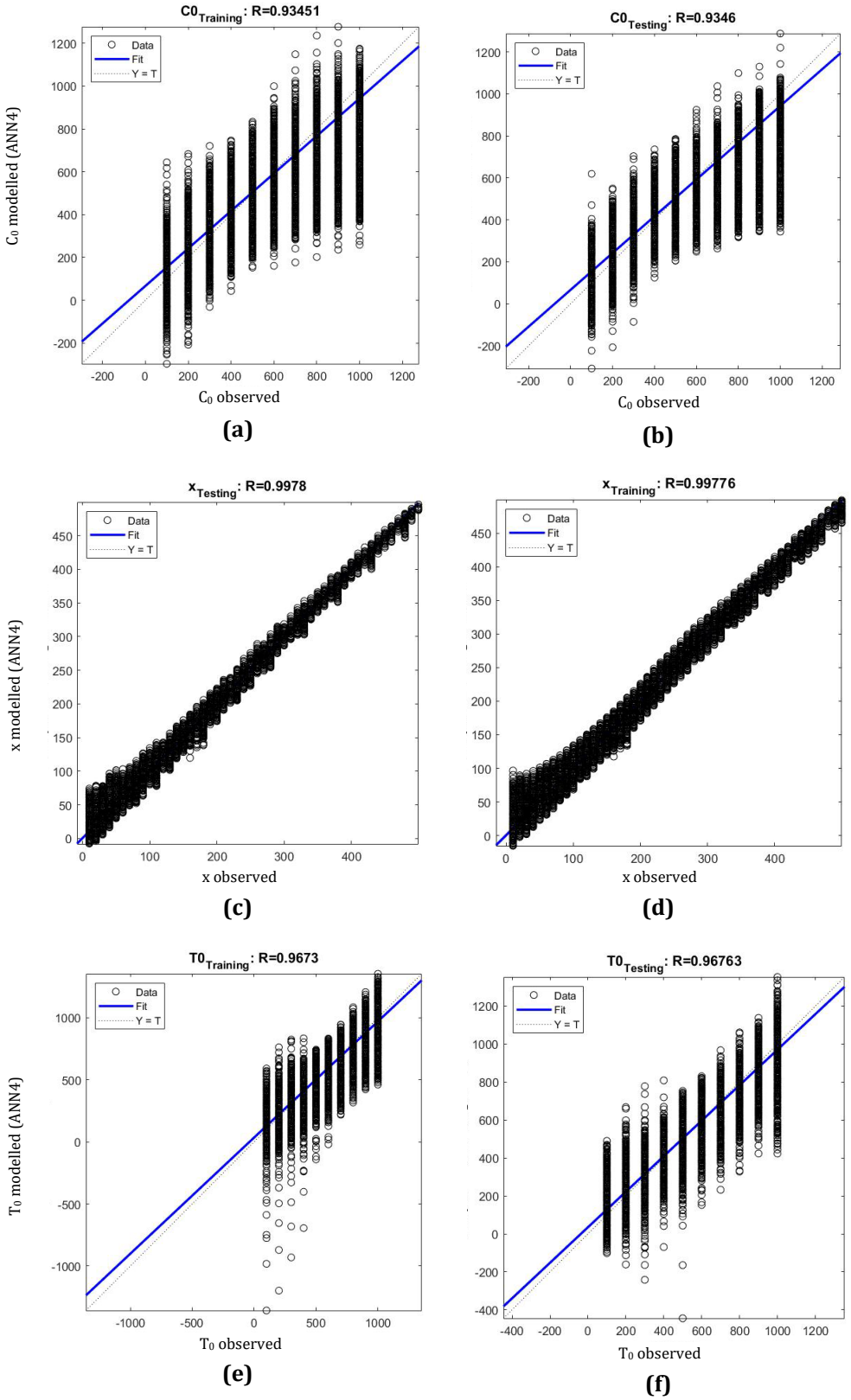

Fig. 7 *Optimal ANN4 model architecture regression plots for (a) C_0 in training; and (b) C_0 in testing; (c) x in training; and (d) x in testing; (e) T_0 in training; and (f) T_0 in testing*

4. Conclusion

This study focuses on the simultaneous detection of various GPSI properties (source location, concentration flux released at source and release duration) that affects the groundwater contamination from a continuous source, by employing a multi-output ANN model. For single-output ANN models (individual identification), the input

datasets combined various parameters (GPSI, flow, and transport parameters) with eleven discrete values from the breakthrough curve. In contrast, for multiple-output ANN model (simultaneous identification), the inputs included only flow and transport parameters along with eleven discrete breakthrough curve values. Two different training algorithms i.e., Levenberg-Marquardt backpropagation and Bayesian regularization algorithm were used for single output models and based on performance better training algorithm (Bayesian Regularization) was used in multiple (three) output model. The performances of the models were compared based on various statistical parameters.

A network with architecture 16-23-1 was found to be optimum for ANN1 and resulted in AARE of 0.56 for identifying the source release concentration. A network with architecture 16-21-1 was found to be optimum for ANN2 and resulted in AARE of 0.26 for identifying the source location. A network with architecture 16-23-1 was found to be optimum for ANN3 and resulted in AARE of 0.51 for identifying the duration of pollutant release. For ANN4 model, network architecture 14-15-3 was found to be optimum and resulted in an average absolute relative error (AARE) of 9.17, 19.96 and 16.75 for identifying the source location, source release concentration and the duration of pollutant release, simultaneously.

Results revealed that single output ANN models performed better than the multiple outputs ANN model. While comparing for individual source parameters (C_0, x and T_0) as target variables, both the model types (single and multiple outputs) could identify the source location very well. Although the release concentration and duration of pollutant release estimation by single output model outperformed the multiple output model. The higher number of predictands increased the complexity and non-linearity of the problem which the ANN based soft-computation model was able to regenerate moderately. Future work may involve employing more sophisticated deep learning models to achieve higher prediction accuracy. Overall, the proposed solution of the problem considered here, has a practical application in formulating cost-effective strategies for regulating and penalizing the agencies accountable for the groundwater pollution.

Acknowledgement

The authors declare that no funds, grants, or other support were received explicitly for the conduct of this study. The authors are thankful to the editor and the anonymous reviewers for their time and effort, and for their constructive comments and suggestions, that helped the authors improve this manuscript. The Authors are thankful to the Department of Civil Engineering (Geoinformatics lab, Water Resources Division) and Institute Computer Centre (Prabha Bhawan) of the Malaviya National Institute of Technology Jaipur (India) for providing the necessary computational facility. The third author is thankful to Ministry of Human Resources Development (Government of India) for providing Teaching-assistantship to the students for pursuing his post-graduation (M.Tech.) at MNIT Jaipur, during which this paper was finalized. The authors have no relevant financial or non-financial interests to disclose with reference to this study.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design****: Jyoti Chaubey and Himanshu Arora;* ***data generation****: Shreyanshu Saha;* ***analysis and interpretation of results****: Shreyanshu Saha and Himanshu Arora;* ***draft manuscript preparation****: Himanshu Arora, Jyoti Chaubey and Shreyanshu Saha. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] Wagner, B. J. (1992) Simultaneous parameter estimation and contaminant source characterization for coupled groundwater flow and contaminant transport modeling, *Journal of Hydrology, 135(1-4)*, 275–303, https://doi.org/10.1016/0022-1694(92)90092-A

[2] Skaggs, T. H. & Kabala, Z. J. (1994) Recovering the release history of a groundwater contaminant, *Water Resources Research, 30(1)*, 71–79, https://doi.org/10.1029/93WR02656

[3] Skaggs, T. H. & Kabala, Z. J. (1995) Recovering the release history of a groundwater contaminant plume: method of quasi-reversibility, *Water Resources Research, 31(11)*, 2669–2673, https://doi.org/10.1029/95WR02383

[4] Sidauruk, P., Cheng, A. H. D. & Ouazar, D. (1998) Ground water contaminant source and transport parameter identification by correlation coefficient optimization, *Ground Water, 36(2)*, 208–214, https://doi.org/10.1111/j.1745-6584.1998.tb01085.x

[5] Atmadja, J. & Bagtzoglou, A. C. (2001) Pollution source identification in heterogeneous porous media, *Water Resources Research, 37(8)*, 2113-2125, https://doi.org/10.1029/2001WR000223

[6] Bagtzoglou, A. C., Dougherty, D. E. & Tompson, A. F. B. (1992) Application of particle methods to reliable identification of groundwater pollution sources, *Water Resources Management, 6(1)*, 15–23, https://doi.org/10.1007/BF00872184

[7] Woodbury, A. D. & Ulrych, T. J. (1996) Minimum relative entropy inversion: theory and application to recovering the release history of a groundwater contaminant, *Water Resources Research, 32(9)*, 2671–2681, https://doi.org/10.1029/95WR03818

[8] Snodgrass, M. F., & Kitanidis, P. K. (1997) A geostatistical approach to contaminant source identification, *Water Resources Research, 33(4)*, 537–546, https://doi.org/10.1029/96WR03753

[9] Michalak, A. M. & Kitanidis, P. K. (2003) A method for enforcing parameter nonnegativity in bayesian inverse problems with an application to contaminant source identification, *Water Resources Research, 39(2)*, 1033, https://doi.org/10.1029/2002wr001480

[10] Michalak, A. M. & Kitanidis, P. K. (2004) Application of geostatistical inverse modeling to contaminant source identification at Dover AFB, Delaware, *Journal of Hydraulic Resources, 42(Sup1)*, 9–18, https://doi.org/10.1080/00221680409500042

[11] Gorelick, S. M., Evans, B. E. & Remson, I. (1983) Identifying sources of groundwater pollution: An optimization approach, *Water Resources Research, 19(3)*, 779–790, https://doi.org/10.1029/WR019i003p00779

[12] Srivastava, D. & Singh, R. M. (2014) Breakthrough Curves Characterization and Identification of an Unknown Pollution Source in Groundwater System Using an Artificial Neural Network (ANN), *Environmental Forensics, 15(2)*, 175-189, https://doi.org/10.1080/15275922.2014.890142

[13] Prakash, O. & Datta, B. (2014) Characterization of groundwater pollution sources with unknown release time history, *Journal of Water Resource and Protection, 6*, 337-350, https://doi.org/10.4236/jwarp.2014.64036

[14] Chaubey, J. & Kashyap, D. (2017) A data parsimonious model for capturing snapshots of groundwater pollution sources, *Journal of Contaminant Hydrology, 197*, 17-28, https://doi.org/10.1016/j.jconhyd.2016.12.008

[15] Gurarslan, G. & Karahan, H. (2015) Solving inverse problems of groundwater-pollution-source identification using a differential evolution algorithm, *Hydrogeology Journal, 23*, 1109-1119, https://doi.org/10.1007/s10040-015-1256-z

[16] Leichombam, S. & Bhattacharjya, R. K. (2016) Identification of Unknown Groundwater Pollution Sources and Determination of Optimal Well Locations Using ANN-GA Based Simulation-Optimization Model, *Journal of Water Resource and Protection, 8*, 411-424, https://doi.org/10.4236/jwarp.2016.83034

[17] Han, K., Zuo, R., Ni, P., Xue, Z., Xu, D., Wang, J. & Zhang, D. (2020) Application of a genetic algorithm to groundwater pollution source identification, *Journal of Hydrology*, *589*, 125343, https://doi.org/10.1016/j.jhydrol.2020.125343

[18] Zhao, Y., Lu, W. & An, Y. (2015) Surrogate Model-Based Simulation-Optimization Approach for Groundwater Source Identification Problems, *Environmental Forensics, 16(3)*, 296-303, https://doi.org/10.1080/15275922.2015.1059908

[19] Zhao, Y., Lu, W., & Xiao, C. (2016) A Kriging surrogate model coupled in simulation–optimization approach for identifying release history of groundwater sources, *Journal of Contaminant Hydrology, 185-186*, 51-60, https://doi.org/10.1016/j.jconhyd.2016.01.004

[20] Jiang, X., Ma, R., Wang, Y., Gu, W., Lu, W., & Na, J. (2021) Two-stage surrogate model-assisted Bayesian framework for groundwater contaminant source identification, *Journal of Hydrology, 594*, 125955, https://doi.org/10.1016/j.jhydrol.2021.125955

[21] Li, J., Lu, W. & Luo, J. (2021) Groundwater contamination sources identification based on the Long-Short Term Memory network, *Journal of Hydrology, 601*, 126670, https://doi.org/10.1016/j.jhydrol.2021.126670

[22] Pan, Z., Lu, W., Wang, H. & Bai, Y. (2023) Groundwater contaminant source identification based on an ensemble learning search framework associated with an auto xgboost surrogate, *Environmental Modelling and Software*, *159,* 105588, https://doi.org/10.1016/j.envsoft.2022.105588

[23] Kontos, Y. N., Kassandros, T., Perifanos, K., Karampasis, M., Katsifarakis, K. L. & Karatza, K. (2022) Machine learning for groundwater pollution source identification and monitoring network optimization, *Neural Computing and Applications, 34*, 19515–19545, https://doi.org/10.1007/s00521-022-07507-8

[24] Xiong, Y., Luo, J., Liu, X., Liu, Y., Xin, X. & Wang, S. (2022) Machine learning-based optimal design of groundwater pollution monitoring network, *Environmental Research, 211*, 113022, https://doi.org/10.1016/j.envres.2022.113022

[25] Xu, Y., Lu, W., Pan, Z., Luo, C., Bai, Y. & Qiu, S. (2024) Groundwater contaminant source identification considering unknown boundary condition based on an automated machine learning surrogate, *Geoscience Frontiers, 15*, 101732, https://doi.org/10.1016/j.gsf.2023.101732

[26] Singh, R. M. & Datta, B. (2007) Artificial Neural Network Modeling for Identification of Unknown Pollution Sources in Groundwater with Partially Missing Concentration Observation Data, *Water Resources Management, 21*, 557-572, http://dx.doi.org/10.1007/s11269-006-9029-z

[27] Bora, T. & Bhattacharjya, R. K. (2015) Development of Unknown Pollution Source Identification Models Using GMS ANN–Based Simulation Optimization Methodology, *Journal of Hazardous Toxic and Radioactive Waste, 19(3)*, 1-12, https://doi.org/10.1061/(ASCE)HZ.2153-5515.0000242

[28] Secci, D., Molino, L. & Zanini, A. (2022) Contaminant source identification in groundwater by means of artificial neural network, *Journal of Hydrology, 611*, 128003, https://doi.org/10.1016/j.jhydrol.2022.128003

[29] Chaubey, J. & Srivastava, R. (2022) Simultaneous identification of groundwater pollution source location and release concentration using Artificial Neural Network, *Environmental Forensics, 23(3-4)*, 293-300, https://doi.org/10.1080/15275922.2020.1850566

[30] Wilson, J. L. & Liu, J. (1994) Backward tracking to find the source of pollution, *Waste Management from risk to remediation, 1*, 181-199.

[31] Neupauer, R. M. & Wilson, J. L. (1999) Adjoint method for obtaining backward in time location and travel time probabilities of a conservative groundwater contaminant, *Water Resources Research, 35(11)*, 3419-3429, https://doi.org/10.1029/1999WR900190

[32] Kumar, J., Jain, A., & Srivastava, R. (2006) Neural network based solution for locating groundwater pollution sources, *Hydrology Journal,* 55-66

[33] Bashi-Azghadi, S. N., Kerachian, R., Bezargan-Lari, M. & Solouki, K. (2010) Characterizing an unknown pollution source in groundwater resources systems using PSVM and PNN*, Expert Systems with Applications, 37(10)*, 7154–7161, https://doi.org/10.1016/j.eswa.2010.04.019

[34] Ogata, A. & Banks, R. (1961) A solution of the differential equation of longitudinal dispersion in porous media, *US Geographical Survey*.

[35] Troiano, M., Nobile, E., Mangini, F., Mastrogiuseppe, M., Conati Barbaro, C. & Frezza, F. (2024) A Comparative Analysis of the Bayesian Regularization and Levenberg Marquardt Training Algorithms in Neural Networks for Small Datasets: A Metrics Prediction of Neolithic Laminar Artefacts, *Information, 15*, 270, https://doi.org/10.3390/ info15050270