VILNIUS TECH
Vilnius Gediminas Technical University

Journal of Environmental Engineering and Landscape Management
ISSN 1648–6897 / eISSN 1822-4199
2022 Volume 30 Issue 4: 515–526
https://doi.org/10.3846/jeelm.2022.18059

APPLICATION NEURAL NETWORK APPROACH FOR THE ESTIMATION OF HEAVY METAL CONCENTRATIONS IN THE INAOUEN WATERSHED

Rachid EL CHAAL*, Moulay Othman ABOUTAFAIL

ENSA of Kenitra, Engineering Sciences Laboratory, Data Analysis, Mathematical Modeling and Optimization Team, Ibn Tofail University, Kenitra, Morocco

Received 19 December 2021; accepted 09 August 2022

Highlights

- The ANN model was developed to predict the heavy metal.
- The sum of the squared errors (SSE), and the correlation coefficient (R) was used to evaluate the performance of the models.
- BFGS algorithm are very powerful tools for training the ANN model.
- The relationships between the parameter physicochemical and heavy metal are constructed.
- The Back Propagation neural network is more accurate in predicting the heavy metal.

Abstract. This paper describes how the multilayer perceptron neural network (MLPNN) trained by the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-newton back-propagation approach was used to estimate heavy metal concentrations: Aluminum (Al), Lead (Pb), Copper (Cu), and Iron (Fe), in the province of Taza using sixteen physicochemical factors measured from 100 samples collected from surface water sources by our team, according to the protocol of the national water office (ONE). We chose a network with only one hidden layer to identify the network architecture to employ. The number of neurons in the hidden layer was varied, as were the types of transfer and activation functions, and the BFGS learning method was used. The following statistical metrics were used to evaluate the performance of the neural network's stochastic models: Examining the adjustment graphs and residue, as well as the Error Sum of Squares (SSE); the mean bias error (MBE) and determination coefficient (R^2). The results reveal that the predictive models created using the artificial neural network method (ANN) are quite efficient, thanks to the BFGS algorithm's efficiency and speed of convergence. An architectural network [16-8-1] (16: number of variables in input layer, 8: number of hidden layer, 1: number of variables in output layer) produced the best results,{R^2: Al(0.954), Pb(0.942), Cu(0.921), Fe(0.968)}, {SSE: Al(0.396), Pb(0.0059), Cu(0.252), Fe(4.29)} and {MBE: Al(–0.033), Pb(0.008), Cu(–0.004), Fe(0.091)}, which is developed so that each model is responsible for estimating the concentration of a single heavy metal. This result demonstrates that there is a non-linear relationship between the physical-chemical properties evaluated and the heavy metal content of surface water in the Taza province.

Keywords: algorithm BFGS, neural networks, heavy metals, environmental processes modelling, prediction.

Introduction

Water quality is being impacted by industrial, agricultural, and urban growth, making it unsafe to drink. This is the case in the Oued Inaouen basin, which is undergoing diversification and seeing an increase in the number of pollutants discharged into the aquatic environment without treatment. Chemical pollution, particularly heavy metal contamination, can occur depending on the source of the waste. The concentrations of these metals can be predicted using strong statistical algorithms based on a set of physicochemical characteristics. Artificial neural networks (ANN) are one of the computer modelling methods utilized in this context. In the modelling and forecasting of environmental characteristics, neural networks have proven to be quite successful.

Artificial neural networks are networks of elementary processors that operate in parallel and are closely coupled. Based on the data it receives, each elementary processor generates a unique output (Touzet, 2016). Because it is an

*Corresponding author. E-mail: *rachid.elchaal@uit.ac.ma* effective process of complicated approximation systems that are difficult to represent using stochastic approaches, neural networks have become a frequently utilized tool in numerous domains (physical, biological, environmental, hydrological, etc.) (Coulibaly et al., 1999). The multilayer perceptron (MLP) type ANN was chosen for our research because it has shown to be effective in predicting hydrological events to avoid the complication of the network and minimize the computation and convergence time of the learning algorithm.

The first goal of this article is to describe the type of ANN architecture chosen and its learning algorithm to comprehend its mathematical concept and operating mechanism, as well as to introduce the BFGS (Broyden, Fletcher, Goldfarb, Shanno) quasi-Newton type algorithm, which has a quadratic form for this network's training, which allows for global convergence in the opposite direction of the descending gradient (Antoine et al., 2007).

The second goal is to forecast the levels of four heavy metals in Taza's surface waters: Al, Cu, Fe, and Pb, utilizing 16 parameters and an environmental database, as well as an MLP-type ANN method using a BFGS learning algorithm. Physical-chemical evaluation of the models developed and assessment of the BFGS algorithm's efficiency and speed of convergence on this sort of data. STATISTICA (version 12.5.192.5 year 4 June 2014) software is used to implement the ANN models.

Then we may define our prediction problem (predicting heavy metals) as a cost function optimization problem (the squared error). By using a back-propagation technique, the goal is to minimize this function by satisfying the problem's constraints (explanatory variables). The connection weights are updated using stochastic gradient descent techniques (Jacobs, 1988). We're interested in MLP multilayer neural networks because of their efficiency and universal approximation properties, as demonstrated by some studies. Cybenko (1989) and Funahashi (1989) and improved by (Hornik et al., 1989), which allows any sufficiently regular bound-ed function to be uniformly approximated with arbitrary precision in a finite domain of the space of its variables by a neural network consisting of an infinite number of hidden neurons, all of which have the same activation function, and a linear output neuron (Hornik et al., 1989; Hornik, 1991).

Various Artificial Intelligence (AI) strategies in the modelling environment have been researched by several authors, including:

Anagu et al. (2009) used artificial neural networks (ANN) to construct models to estimate heavy metal sorption in German soils based on basic soil parameters. El Badaoui et al. (2012) used artificial neural networks and multiple linear regressions using physical-chemicals characteristics to construct stochastic mathematical models for the prediction of heavy metal levels in surface waters of the Oued Boufekrane watershed (Meknes, Morocco). Abdallaoui and El Badaoui (2015) compared the predictive capacity of MLR and ANN for estimating the concentration of four heavy metals (Cd, Cr, Cu, and Pb) from eight different physical chemicals (organic matter, content in water, fine fraction, pH, $CaCO_3$, carbon and phosphorus in sediments, and suspended matter in the water column) in the Beht river basin in Morocco. Bayatzadeh Fard et al. (2017) employed biogeography-based optimization (ANN-BBO) and adaptive neural fuzzy inference system multiple outputs (MANFIS) to predict heavy metal distribution in groundwater at the Lakan lead-zinc mine in Iran. Verma and Singh (2013) used the Artificial Neural Network (ANN) approach using a dataset generated from the analysis of mine water discharge from the Jharia coal basin in Jharkhand, India, to calculate BOD and COD from factors such as temperature, pH, DO, TSS. Shouyu and Honglan (2005), developed a fuzzy optimization neural network strategy for predicting the frost date and rupture date in the Inner Mongolia part of the Yellow River using hydro-meteorological data.

The goals of this study were to develop a relevant stochastic mathematical model for the prediction of iron, aluminium, copper, and lead levels, based on MLPNN, from some physicochemical parameters, in the Inaouen catchment area of the TAZA region to address the problem of heavy metal pollution of superficial waters and to provide better management of water quality in providing an adequate water supply. Because of their basic architecture, their capacity to execute arbitrary transformations from inputs to outputs, and their usage in a wide range of prediction problems.

The model was developed using data acquired by our team between 2014 and 2015 from 100 sources in the Inaouen watershed, which were collected according to the National Drinking Water Office's methodology (ONEP).

1. Material and methods

Description of the study area: Taza is a mostly rural subdivision of the Fez-Meknes region, located in Morocco's north-eastern area (Figure 1). Taza province has a humid subtropical climate. The study area's rainfall regime is divided into two seasons: a rainy season that lasts about eight months, from October to May, with an average rainfall of around 580 mm, and a dry season that lasts from June to September, with average maximum temperatures nearing 35 °C. In terms of morphology, the Taza region is wedged between two hydrological basins: the Moulouya to the east and the Sebou to the west (El Haji et al., 2012). The main river Oued Inaouen and its tributaries (Oued Larbâa, Oued Bouhlou, Oued Lahder, and Oued Matmata) cut through the province of Taza, creating a dense hydrographic network (Figure 2). The province of Taza has two types of water tables: a deep water table and a water table, both of which are hydrogeological significant. The province has a unitary sanitation network. It accepts wastewater and rainwater through the same pipelines and discharges them into the natural environment without treatment (Ben Abbou et al., 2013).

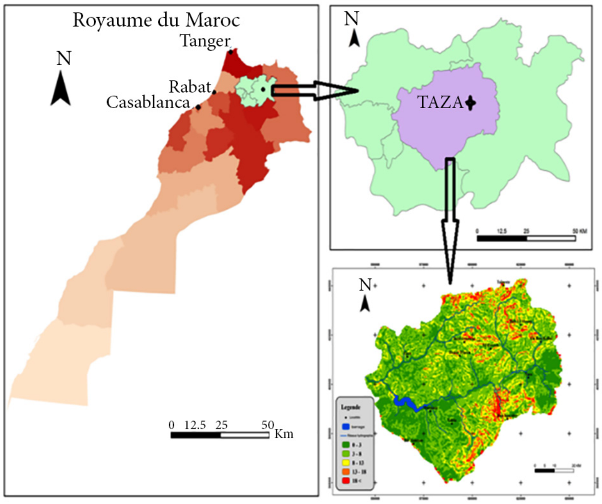

Figure 1. Geographical location of the study area

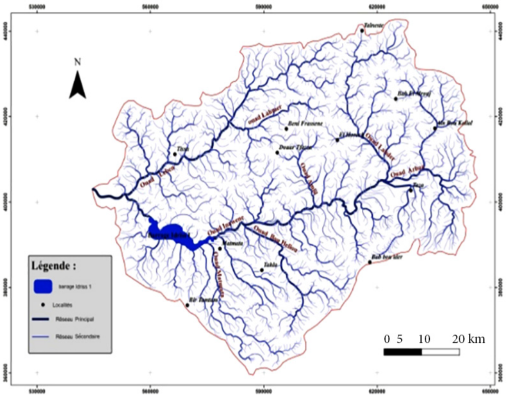

Figure 2. Map of the hydrographic network of the study area

Database: Our database contains 100 surface water samples (observations) collected in the province of Taza between 2014 and 2015; water samples were collected, transported, and stored according to the National Drinking Water Office's (ONEP) policy and procedures. A portion of the analysis was done there, and another portion was done at the Regional University Centre of Interface (CURI) attached to the Sidi Mohamed Ben Abdellah University (USMBA) of Fez at the time of sampling to avoid the change and temperature, conductivity, pH using a multi-parameter analyzer Type CONSORT – Model C535, and dissolved oxygen by the titration method of Winkler. The methods used at CURI Laboratory: volumetry for bicarbonates, chlorides, calcium and magnesium; molecular absorption spectrophotometry for sulfates, nitrates, nitrites, ammonium ions and orthophosphates; and flame spectrophotometry for sodium and potassium. The analysis of the four heavy metals is performed by atomic absorption spectrum (AAS).

Selection of inputs: The sixteen independent (explanatory) variables are the Temperature (T °C) and physical-chemical parameters determined in these samples: hydrogen potential (pH) (the concentration of H^+ ions in the water), dissolved oxygen O_2 (Oxy Diss), Sodium (Na^+), total dissolved solids (TDS)), Conductivity (Cond), Bicarbonate (HCO_3^-), Magnesium (Mg^{2+}), Total Alkalinity (as $CaCO_3$), Chlorides (Cl^-), Potassium (K^+), Nitrate (NO_3^-), Calcium (Ca^{2+}), Sulfates (SO_4^{2-}), Ammoniac (NH_4^+), and Phosphorus (P).

The heavy metal contents are the dependent variables (to be forecasted), and there are four of them: Iron (Fe^{2+}), Aluminium (Al^{3+}), Copper (Cu^{2+}), and Lead (Pb^{2+}).

Selecting the percentage of data for the proposed training often involves trial and error; the researchers

Table 1. The statistical summary of data

Parameter	Number of data	Mean	Median	Minimum	Maximum	Variance	Standard Deviation
T °C	100	19.90	19.50	16.30	23.00	3.54	1.88
pH	100	7.19	7.03	5.21	9.61	1.18	1.09
Oxy Diss	100	5.04	4.64	0.00	14.01	12.71	3.56
Cond	100	233.69	130.50	0.00	2959.00	140 492.05	374.82
TDS	100	117.15	66.00	0.00	1480.00	35237.44	187.72
HCO_3^-	100	80.63	57.95	3.66	634.40	7788.82	88.25
$CaCO_3$	100	66,09	47.50	3.00	520.00	5233.01	72.34
Mg^{2+}	100	3.17	2.05	0.20	25.00	14.48	3.81
Na^+	100	24.51	5.40	1.00	540.00	4197.42	64.79
K^+	100	1.72	0.97	0.10	13.00	4.43	2.10
Cl^-	100	25.84	2.45	0.40	550.00	5676.23	75.34
Ca^{2+}	100	17.79	15.00	1.00	110.00	316.65	17.79
SO_4^{2-}	100	7.83	5.00	0.28	150.00	244.37	15.63
NO_3^-	100	0.58	0.41	0.03	3.20	0.29	0.53
P	100	32.61	30.00	10.00	80.00	196.52	14.02
NH_4^+	100	27.30	25.00	10.00	80.00	210.19	14.50

have chosen between 70 and 80% of the data for training (Rahnama et al., 2020). The following is how the database is distributed: 70% of the samples for the learning phase of a predictive model of the predictor variables were randomly selected from the entire dataset. The remaining 30% of the samples were used to double-check network performance during training and prevent over-learning. This is to see if the predictions made by these models are accurate and reliable (15 per cent for testing and 15 per cent for the validation process).

Formatting of the data: (Abdallaoui & El Badaoui, 2015; Bayatzadeh Fard et al., 2017; Patro & sahu, 2015).

Standardization is a data preprocessing technique that helps to reduce model complexity. The input data consists of raw, unprocessed values (16 independent variables). Their scales are very dissimilar. These data are turned into standardized variables to balance the measurement scales. Indeed, the values of each independent variable I have been normalized in terms of their means and standard deviation by the relationship:

$$X_s(v_i) = \left(\frac{X(v_i) - \overline{X}(v_i)}{\sigma(v_i)} \right), \; i \in \{1 \ldots 16\}, \tag{1}$$

where: $X_s(v_i)$ – normalized value for the variable x_i; $X(v_i)$ – value observed for the variable i; $\overline{X}(v_i)$ – mean value for the variable i.

$$\overline{X}(v_i) = \frac{1}{100} \sum_{k=1}^{100} X_k(v_i); \tag{2}$$

$\sigma(v_i)$ – standard deviation for the variable i,

$$\sigma(v_i) = \sqrt{\frac{1}{100} \sum_{k=1}^{10} (X_k(v_i) - \overline{X}(v_i))^2}. \tag{3}$$

This transformation assigns to the variable a majority of values between [–1, 1], the result being less dependent on outliers. Then the goal of normalizing all parameter values is to avoid very large or very small exponential calculations and to keep variation at a minimum. To match the needs of the transfer function utilized by neural networks, the values corresponding to the explanatory variable were additionally standardized in the interval [0; 1]. This normalizing was carried out using the following formula:

$$\mathbf{Y}_n = \left(\frac{Y - Y_{\min}}{Y_{\max} - Y_{\min}} \right), \tag{4}$$

where: $\mathbf{Y}_n$ – normalized value; Y – observed value; $Y_{\min}$ – minimum value; $Y_{\max}$ – maximum value.

Multilayer Perceptron (MLP): A simplified model of a biological neuron is used to create the artificial neuron. As a result, ANN is a mathematical modelling technique that mimics the behaviour of organic neurons, as described by Mc Culloch and Pitts (Touzet, 2016). A neural network's formal neuron is its most important component. It is a mathematical operator that receives some external signals from the outputs of other neurons as input variables, each of which is connected with weights. Figure 3 shows the single output. It is a non-linear function φ of a linear combination A_j (also called the potential) of the inputs (X_n). The combination function A_j the most frequently used is the weighted sum of the inputs (X_n) weighted by the coefficients (W_n) representative of the connection strength, also called connection weight (El Hmaidi et al., 2013; Skansi, 2018):

$$A_j = B_j + \sum_{i=1}^{16} W_{ji} X_i. \tag{5}$$

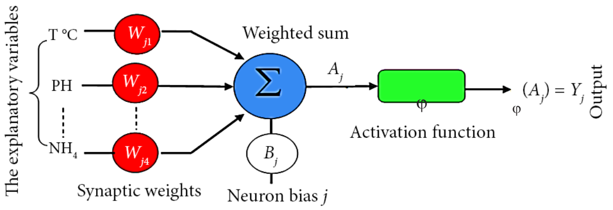

Figure 3. Structure of a formal neuron

Or B_j is the bias of neuron j, W_{ij} is the synaptic weight and X_i is the input value relative to the variable i. And we can also write it in matrix form (Graupe, 2007):

$$A_j = B_j + W_j^T X, \tag{6}$$

with: $W_j^T = (W_{j1} \ldots W_{jn})$ and $X^T = \left(X_1 \ldots X_n\right)$, and W_j^T being the transposition of W_j.

The output of neuron j is calculated according to the following formula (Parizeau, 2004; Skansi, 2018; Touzet, 2016):

$$Y_j = \varphi_j\left(A_j\right) = \varphi_j\left(B_j + \sum_{i=1}^{16} W_{ji} X_i \right) = \varphi_j\left(B_j + W_j^T X\right). \tag{7}$$

The hidden layer's neuron j has φ_j the function of activation or transfer (Figure 4).

The output equation in a three-layer MLP (16 neurons in the input layer, n neurons in the hidden layer, and a single neuron in the output layer) is as follows (Figure 5) (n = 8):

$$Y_{prédi} = f\left(\sum_{j=1}^{n} W_j \times \varphi_j \left(B_j + \sum_{i=1}^{16} W_{ji} X_i \right) \right) + B. \tag{8}$$

Or $Y_{prédi}$ is the output of the output layer; f and φ_j are respectively the activation functions of the neuron of the output layer and the neuron j of the hidden layer; W_j and W_{ij} are respectively the weight between the $j^{ème}$ neuron and the output neuron and the weight between the $i^{éme}$ neuron and the $j^{ème}$ neuron and B is the bias of the output layer (Kharroubi et al., 2016).

Most of the activation functions used for the neurons of the hidden layer are: continuous, non-linear, differentiable (which allows it to create optimizations based on

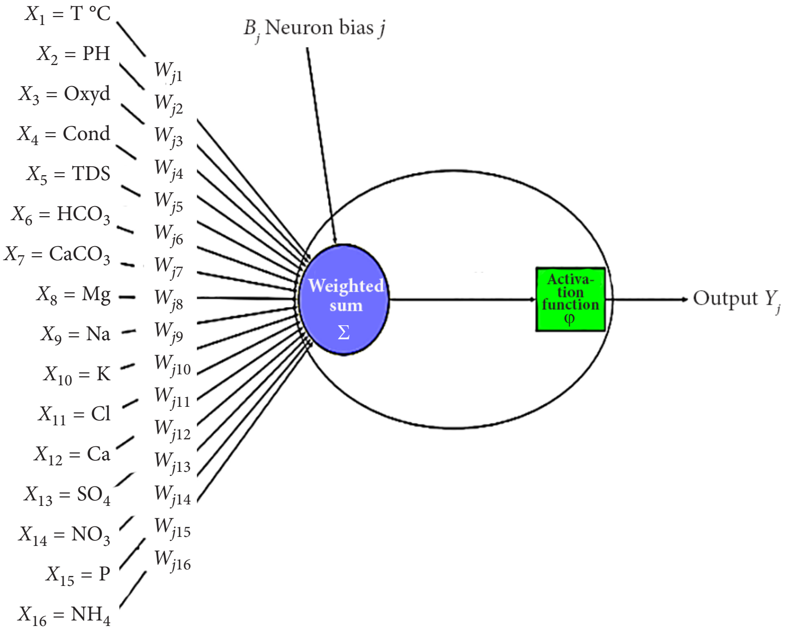

Figure 4. Model of a formal neuron with the sixteen Physico-chemical parameters as inputs

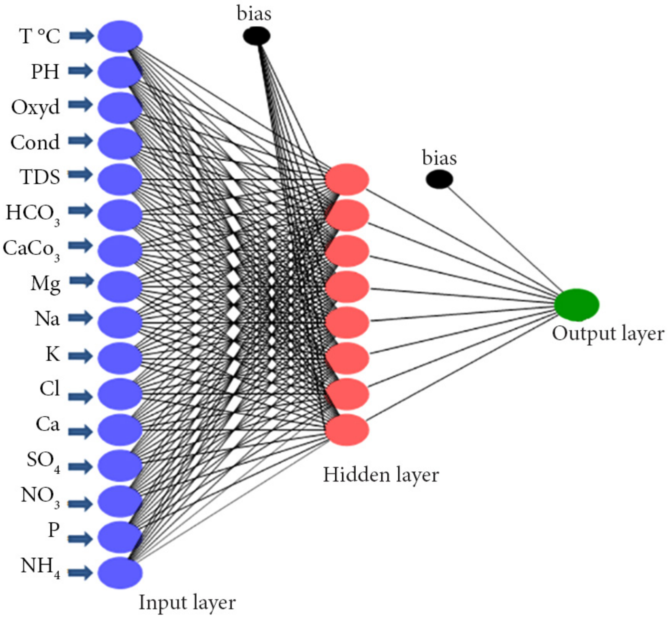

Figure 5. Architecture of the neural network with three configuration layers [16-8-1] used in this study

the gradient method), and bounded as sigmoid, hyperbolic tangent, Arctangent, Heaviside, Gaussian function (Parizeau, 2004; Snyman & Wilke, 2018). For the output layer neuron, the function that calculates the output of the network based on its activation state; in general, it is the identity function (Chamekh, 2007).

The activation functions used in our work are the hyperbolic tangent, logistic sigmoid, sine, exponential and identity functions:

- The linear activation function or identity function. The function does nothing with the input's weighted sum; it simply returns the value it was given: $f(x) = x$.
- Sigmoid / Logistic Activation Function: This function accepts any real value as an input and outputs values between 0 and 1: $f(x) = \frac{1}{1+e^{-x}}$.
- Tanh Function (Hyperbolic Tangent): Tanh function is very similar to the sigmoid/logistic activation function and even has the same S-shape, with the exception that the output range is –1 to 1: $f(x) = \frac{(e^{x} - e^{-x})}{(e^{x} + e^{-x})}$.
- The sine function: $f(x) = \sin(x)$.
- The negative exponential function: $f(x) = e^{-x}$.

Network architecture and learning rule: A neural network is a network of neurons that is often structured in layers. The ideal MLP model architecture and implementation boil down to selecting transfer functions, finding important inputs, deciding the number of neurons in the hidden layer, selecting a learning method, and finally optimizing and testing the network (Boudad et al., 2015). I can mention that it is possible to predict the four heavy metals at the same time with a single MLPNN model whose output layer consists of four neurons referring to the four heavy metals; this approach can be the subject of another future work or an extension of this modest work. Several works of literature have shown the possibility of simulating multi-input and multi-output systems using ANNs (Karimi et al., 2021).

Implementation under STATISTICA 12: By studying the different papers on water resources, it appeared that the structure of the hidden layer with a variable number of neurons is more appropriate and gives better results (Hornik et al., 1989). Based on the "universal approximator theory" by Hornik et al. (1989), all the applied networks are made of a hidden layer. This theory allows reducing to the minimum the number of hidden layers (Asadollahfardi et al., 2012).

Simple single-layer networks are substantially less powerful than multilayer perceptron (MLP) networks. There are three layers in the network employed in this study: an input layer, a hidden layer, and an output layer. The buried layer's number of neurons and activation functions are not predetermined. They are formed during the learning process. STATISTICA 12 statistics software was used to run the neural network simulations, because it has an efficient tool to search the best MLP network architecture, called ANS: automated network search.

The implementation of multilayer neural networks has two parts of the design: determining the architecture of the network and an optimization numerical calculation part. This calculation is the determination of the synaptic coefficients and the updating of these coefficients by a supervised learning algorithm. The algorithm chosen for our study is the BFGS algorithm. Because STATISTICA 12 integrates some high-performance learning algorithms, notably Conjugate Gradient Descent, BFGS, and besides STATISTICA 12, there are other algorithms such as Levenberg-Marquardt, which are used in several research works such as (Asadollahfardi et al., 2018; Zhou et al., 2021; Davoudi & Vaferi, 2018; Vaferi et al., 2011). A comparative study between the two algorithms BFGS and Levenberg-Marquardt is the subject of my recently published work (El Chaal & Aboutafail, 2022). Algorithm BFGS seeks to minimize, by non-linear optimization methods, a cost function (sum of the squares errors of prediction (SSE)) which constitutes a measure of the difference between the actual responses of the network and its desired responses. This optimization is done iteratively by modifying the weights as a function of the gradient of the cost function: the gradient is estimated by a specific method for neural networks, called the back-propagation method, and then it is used by the algorithm optimization (Dreyfus, 2006). The weights are randomly initialized before learning.

Evaluation of performances: STATISTICA 12 employs the cost function, sometimes known as the least-squares criteria, which entails reducing the sum of the squares of the residuals. In this example, the network will learn a discriminant function for scoring and evaluating the quality of our predictive model. The sum of the differences between the target values and the expected outputs defined for the training set is the sum of squares error. The outcome of the evaluation can be expressed in two ways: graph analysis and statistical indicators (Boudebbouz et al., 2016). The indicators used in this study are the determination coefficient ($\mathbf{R}^2$) and the squared sum of errors (SSE), which are defined as follows:

The determination coefficient ($\mathbf{R}^2$):

$$\mathbf{R}^2 = \left(\frac{\sum_{i=1}^{100}\left(Y_i^{obs} - \overline{Y^{obs}}\right)\left(Y_i^{prédi} - \overline{Y^{prédi}}\right)}{\sqrt{\sum_{i=1}^{100}\left(Y_i^{obs} - \overline{Y^{obs}}\right)^2 \left(Y_i^{prédi} - \overline{Y^{prédi}}\right)^2}} \right)^2 . \quad (9)$$

The quadratic sum of errors (SSE) (SOS: Sum of Square Error for STATISTICA12):

$$SSE = E(W) = \sum_{i=1}^{100}\left(Y_i^{obs} - Y_i^{prédi}\right)^2 . \quad (10)$$

With $E(W)$ is the function of the cost, which depends on the vector of parameters of the network W (connection weights) (Skansi, 2018) and Y^{obs} is the observed (real) value of the studied metals, $Y_i^{prédi}$ is the estimated value of metals by the model at observation i. $\overline{Y}$ is the mean value. The best prediction is when $|\mathbf{R}|$, on the one hand, SSE, on the other hand, tend toward 1 and 0, respectively (Abdallaoui & El Badaoui, 2015; Bayatzadeh Fard et al., 2017; Boudad et al., 2015).

Optimization algorithm: The BFGS method: For tackling unconstrained optimization problems, quasi-Newton approaches are well-known. Davidon invented these approaches, which use the updated formulas for the Hessian approximation, in 1959, and Fletcher and Powell popularized them in 1963 as the Davidon-Fletcher-Powel (DFP) method. However, the DFP approach is no longer widely employed. Broyden, Fletcher, Goldfarb, and Shanno, on the other hand, proposed the idea of a new updating formula, known as BFGS, in 1970. It has since become extensively utilized and has undergone numerous changes (Ibrahim et al., 2014). The BFGS algorithm should not be parameterized by a learning rate for the training of a neural network.

The following is a general description of unconstrained optimization problems: Is $f \in C^2\left(\mathbb{R}^n, \mathbb{R}\right)$ and $g = \nabla f \in C^1\left(\mathbb{R}^n, \mathbb{R}^n\right)$. We have in this case:

$$f(\mathbf{x}) = \inf_{\mathbb{R}^n} f \Rightarrow g(\mathbf{x}) = 0. \quad (11)$$

If more f is convex than we have $g(\mathbf{x})=0 \Rightarrow f(\mathbf{x})=\inf_{IR^n} f$. In this case of equivalence, we can use Newton's method to minimize f by applying Newton's algorithm to find a zero of $g=\nabla f$. We have $D(\nabla f)=H$ H or is the Hessian matrix of f (Berhail, 2016).

Newton's method is written in this case (Fletcher, 1981):

$$\begin{cases} \text{Initialisation} & x_0 \in \mathbb{R}^n \\ \text{Itération } k & \begin{array}{l} H_K\left(x_{k+1}-x_k\right)= \\ H\left(x_k\right)\left(x_{k+1}-x_k\right)=-\nabla f\left(x_k\right)=-g_k \end{array} \end{cases} . \qquad (12)$$

The quasi-Newton method consists in approaching the inverse of the Hessian; it is of the type (Mottelet, 2003):

$$\begin{cases} d_k & =-H_k^{-1} g_k \\ x_{k+1} & =x_k+\rho_k d_k \end{cases} \qquad (13)$$

or

$$\begin{cases} d_k & =-B_k g_k \\ x_{k+1} & =x_k+\rho_k d_k \end{cases} \qquad (14)$$

or B_k is a matrix intended to approximate the reverse of the Hessian matrix of f in x_k d_k and the direction of travel and ρ_k the optimal step. The strategy adopted is to make this approximation and update the approximation B_k during iterations. We pose $B_0=I$, the idea as follows: we know that at the point x_k, the gradient and the Hessian of f verifying the relation (the limited expansion of g in the neighbourhood of x_{k+1} and applied in x_k (Mottelet, 2003):

$$\begin{array}{l} g_{k+1}=g_k+\nabla^2 f\left(x_k\right)\left(x_{k+1}-x_k\right)+ \\ \in\left(x_{k+1}-x_k\right). \end{array} \qquad (15)$$

If we assume that the quadratic approximation is good, we can then neglect the rest and consider that we have:

$$g_{k+1}-g_k \approx \nabla^2 f\left(x_k\right)\left(x_{k+1}-x_k\right). \qquad (16)$$

This leads to the notion of quasi-Newton relation: We say that the matrix B_{k+1} and H_{k+1} verify a quasi-Newton relation if we have

$$\begin{array}{l} H_{k+1}\left(x_{k+1}-x_k\right)= \\ \nabla f\left(x_{k+1}\right)-\nabla f\left(x_k\right)=g_{k+1}-g_k; \end{array} \qquad (17)$$

or

$$\begin{array}{l} x_{k+1}-x_k= \\ B_{k+1}\left(\nabla f\left(x_{k+1}\right)-\nabla f\left(x_k\right)\right)=B_{k+1}\left(g_{k+1}-g_k\right). \end{array} \qquad (18)$$

It remains updated B_k while maintained $B_k>0$. The BFGS method (from Broyden, Fletcher, Goldfarb and Shanno) seeks to construct B_{K+1} near B_K (minimize the distance between B_{K+1} and B_K), such as B_{K+1} satisfies (3.14) and such that it B_K is symmetric de fi ned positive then B_{K+1} is symmetric de fi ned positive. We pose $s_k = x_{k+1}-x_k=\rho_k d_k$, and we suppose that $s_k \neq 0$, and we pose $\mathbf{y}_k=\nabla f\left(x_{k+1}\right)-\nabla f\left(x_k\right)=g_{k+1}-g_k$.

With a suitable choice of the standard $\mathcal{M}_n(IR)$, we get the approximation of B_{K+1} or H_{k+1} [if $s_k \neq 0$ and $\nabla f\left(\mathbf{x}_k\right) \neq 0$, otherwise, the algorithm stops] that we give here without further details (Fletcher, 1981; Mottelet, 2003; Snyman & Wilke, 2018):

$$B_{k+1}=\left(I-\frac{s_k y_k^\top}{y_k^\top s_k}\right)^\top B_k\left(I-\frac{s_k y_k^\top}{y_k^\top s_k}\right)+\frac{s_k s_k^\top}{y_k^\top s_k}; \qquad (19)$$

or

$$H_{k+1}=H_k+\frac{y_k y_k^\top}{y_k^\top s_k}-\frac{H_k s_k s_k^\top H_k}{s_k^\top H_k s_k} \quad \text{with } H_k=B_k^{-1}. \qquad (20)$$

For our neural network, we have $f(x)=E(W)$, and $x_k=W_K$ with W_K weight matrix at iteration *k*; therefore, the modification of the connection weight matrix (update of the weights during learning) is done according to the following equation:

$$\begin{array}{l} W_{k+1}=W_k-\rho_K H_k^{-1}\dfrac{\partial E(W_K)}{\partial W_k}= \\ W_k-\rho_K B_K\dfrac{\partial E(W_K)}{\partial W_k}=W_k-\rho_K B_K \nabla E(W_K). \end{array} \qquad (21)$$

BFGS algorithm for fitting the vector of weights (Ibrahim et al., 2014): Initialize $k=0$ (k denotes the number of iterations). Randomly initialize the initial weight vector W_0, and we take $H_0=I$ with *I*: the identity matrix.

Will give ε precision a low positive value ($\varepsilon \in \mathbb{R}_+^*$) and $\rho_0=1$.

Calculate the gradient of the error $\nabla E\left(W_k\right)$.

Test if $\left\|\nabla E\left(W_k\right)\right\| \leq \varepsilon$ (if the result of the test corresponds to the answer YES, stop. Otherwise, continue).

Calculate search direction $d_k=-H_K^{-1}\nabla E\left(W_k\right)$.

Wolfe linear search to calculate a step $s_k > 0$.

Adjust the weight vector $W_{k+1}=W_k+\rho_K d_k$.

Update the matrix H_k.

$$H_{k+1}=H_k+\frac{y_k y_k^\top}{y_k^\top s_k}-\frac{H_k s_k s_k^\top H_k}{s_k^\top H_k s_k} \qquad (22)$$

with

$$y_k=\nabla E\left(W_{k+1}\right)-\nabla E\left(W_k\right) \text{ and } s_k=W_{k+1}-W_k .$$

Increment *k* one unit and repeat from step ii.

2. Results and discussion

2.1. Relevancy between independent variables and the four heavy metals

The correlation matrix, in particular, is used to assess and investigate the amount and direction of a linear relationship between two variables. This is a metric for the degree of linearity between two variables. The range of correlation values is –1 to +1. The correlation value is positive if both variables tend to increase and drop at the same time. The correlation value is negative when one variable grows while the other drops. The greater the linear relationship between the variables, the larger the absolute value of the coefficient. The most well-known methods for performing this type of analysis are Spearman, Pearson, and Kendall (Jiang et al., 2021).

The results of the correlation matrix analysis on the obtained database are shown in Figure 6. This figure indicates that Aluminum correlates: strong and positive with Mg^{2+}, and medium and positive with Conductivity, TDS, HCO_3^-, Ca^{2+}, and $CaCO_3$. For iron, it is well correlated positively with Mg^{2+} and Ca^{2+} and medium correlated with HCO_3 and $CaCO_3$. Copper has a strong positive correlation with Conductivity, TDS, HCO_3^-, $CaCO_3$, Mg^{2+}, Na^+, K^+, Cl^- and SO_4^{2-}, for Lead, it is directly related to NO_3^- (strong and positive correlation).

Table 2. Matrix of correlation for 16 variables independent and four heavy metals

Variable	Correlations (donnéeElchaal)			
	Marked correlations are significant at p < .05000 N = 100			
	Al	Fe	Cu	Pb
T °C	0.027	0.158	0.037	0.192
pH	0.316	0.152	0.160	–0.324
Oxy Diss	–0,331	–0.313	–0.205	0.178
Cond	0.541	0.343	0.793	0.086
TDS	0.542	0.343	0.794	0.085
HCO_3^-	0.535	0.456	0.731	0.003
$CaCO_3$	0.535	0.456	0.731	0.003
Mg^{2+}	0.862	0.722	0.512	0.095
Na^+	0.348	0.086	0.792	0.059
K^+	0.487	0.244	0.947	0.004
Cl^-	0.484	0.254	0.728	0.117
Ca^{2+}	0.676	0.949	0.233	0.122
SO_4^{2-}	0.308	0.127	0.614	–0.014
NO_3^-	0.057	0.072	–0.083	0.745
P	–0.082	0.101	0.020	0.080
NH_4^+	0.161	0.111	–0.008	–0.149

2.2. Result of the best networks

The Automated Network Search (ANS) tool from STATISTICA was used to find the results, which are listed in Table 1. They display the determination coefficients, the method type (BFGS), the activation functions of the two layers, and the number of iterations based on the topology.

From the values of the determination coefficients for learning shown in Tables 3 and 4, the architectures were greatest when the number of neurons in the hidden layer NNH = 8 for all metals. Indeed, the results found with the architecture [16-8-1] are the most satisfied with the maximum determination value, and the minimum of the squared sum of the errors, the activation functions used during this study are logistic (for Al and Fe) and exponential (for Pb and Cu) for the hidden layer and identity for the output layer (Table 5).

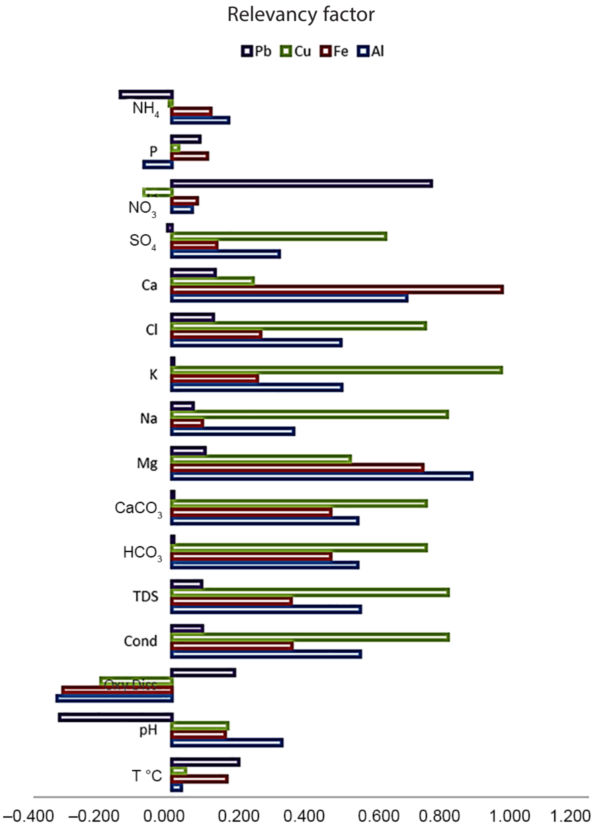

Figure 6. The relevancy factor between independent variables and the four heavy metals

We can add the mean bias error (MBE) to evaluate the performance of the models found by ANN and that if the estimates of a model are statistically significant, it is negative if the model is underestimated, conversely, if it is overestimated.

Table 6 illustrates the statistical results obtained. We note that the models chosen for Pb and Fe overestimate the measurements, and in the model chosen for Al and Cu are overestimated: MBE (Al: –0.033; Pb: 0.008; Cu: –0.004; Fe: 0.091) these indices show the performance of the models found.

$$MBE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i^{prédi} - Y_i^{obs}\right).$$

2.3. Trend lines

Thus, from Figure 7, the observed values and those estimated for the four heavy metals by the neural networks are closer to the trend curve. It shows the performance of this method.

The determination between the observed and estimated heavy metal levels in reference to the models developed by the ANNs technique are depicted in Figure 7. The observed values and the neural network estimates for the four heavy metals are closer to the trend curve.

Table 3. Recap of the five best architectures offered by STATISTICA12 for Al and Pb metal

Result of the best networks							
Metals	Better architecture proposed	**R**^{2}	SSE	Algorithm	Number of iterations	The hidden layer activation function	Layer output activation function
Aluminium (Al)	MLP 16-12-1	0.933	0.584	BFGS	85	Logistic	Logistic
	MLP 16-12-1	0.937	0.539	BFGS	37	Exponential	Tanh
	MLP 16-8-1	0.951	0.418	BFGS	64	Tanh	Sine
	MLP 16-8-1	0.947	0.455	BFGS	53	Logistic	Identity
	MLP 16-8-1	0.955	0.396	BFGS	69	Logistic	Identity
Lead (Pb)	MLP 16-8-1	0.884	0.0126	BFGS	57	Exponential	Logistic
	MLP 16-8-1	0.943	0.0059	BFGS	79	Exponential	Identity
	MLP 16-8-1	0.912	0.0089	BFGS	48	Exponential	Exponential
	MLP 16-8-1	0.906	0.0095	BFGS	52	Exponential	Exponential
	MLP 16-8-1	0.916	0.0085	BFGS	57	Exponential	Exponential

Table 4. Recap of the five best architectures offered by STATISTICA12 for Cu and Fe metal

Result of the best networks							
Metals	Better architecture proposed	**R**^{2}	SSE	Algorithm	Number of iterations	The hidden layer activation function	Layer output activation function
Copper (Cu)	MLP 16-10-1	0.914	0.2778	BFGS	12	Exponential	Identity
	MLP 16-7-1	0.918	0.2686	BFGS	12	Exponential	Identity
	MLP 16-8-1	0.922	0.2527	BFGS	14	Exponential	Identity
	MLP 16-8-1	0.920	0.2657	BFGS	12	Exponential	Identity
	MLP 16-10-1	0.922	0.2545	BFGS	13	Exponential	Identity
Iron (Fe)	MLP 16-7-1	0.964	4.7154	BFGS	62	Logistic	Identity
	MLP 16-11-1	0.955	6.16	BFGS	50	Logistic	Exponential
	MLP 16-8-1	0.968	4.29	BFGS	64	Logistic	Identity
	MLP 16-9-1	0.960	5.45	BFGS	51	Logistic	Exponential
	MLP 16-10-1	0.964	4.70	BFGS	65	Logistic	Identity

Table 5. Determination coefficient **R**^{2} and SSE obtained by STATISTICA12 on the three sets for the best topology of each heavy metal

		Coefficient **R**^{2}			Sum of Squared Errors (SSE)			Function Activation	
Metals	Architecture	Learning	Test	Validation	Learning	Test	Validation	hidden layer	Output layer
Al	MLP 16-8-1	0.955	0.677	0.906	0.396	0.697	0.411	Logistic	Identity
Pb	MLP 16-8-1	0.943	0.880	0.893	0.00592	0.009	0.01937	Exponential	Identity
Cu	MLP 16-8-1	0.922	0.904	0.943	0.252	0.225	0.103	Exponential	Identity
Fe	MLP 16-8-1	0.968	0.945	0.980	4.29	8.54	4.68	Logistic	Identity

Table 6. MBE for the best topology of each heavy metal

Mean Bias Error (MBE)					
Metals	Architecture	Learning	Test	Validation	All data
Al	MLP 16-8-1	0.022	–0.299	–0.027	–0.033
Pb	MLP 16-8-1	0.01	–0.014	0.017	0.008
Cu	MLP 16-8-1	–0.024	0.202	–0.113	–0.004
Fe	MLP 16-8-1	–0.017	–1.191	1.875	0.091

They demonstrate the effectiveness of the ANN technique according to the significance of the determination coefficient, which was 0.92 for copper, 0.95 for aluminium, 0.94 forlead, and 0.96 for iron. If you look at Figure 7, you'll see what I mean. Furthermore, the improved performance of the models established by the principle of ANN based on the BFGS algorithm is once again justified by these results.

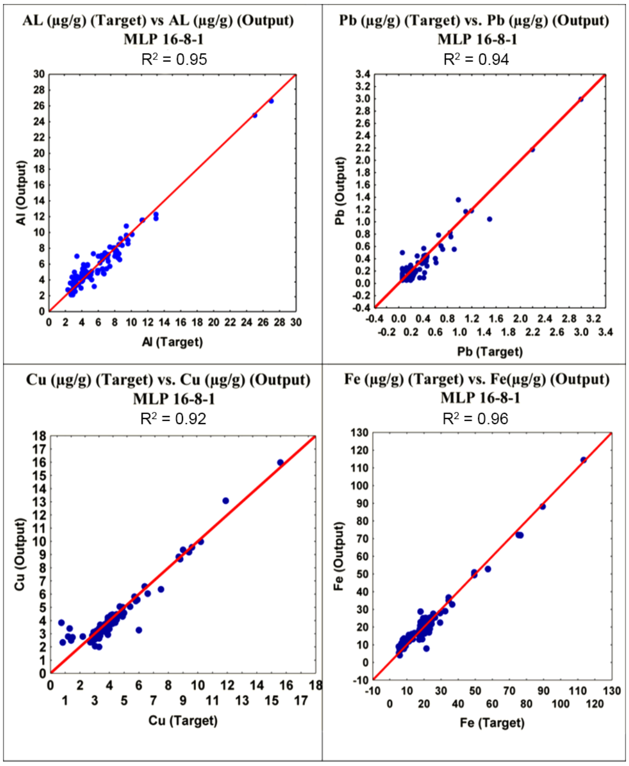

Figure 7. Shows the relation between the observed values and the ANN model's estimated values for the four heavy metals

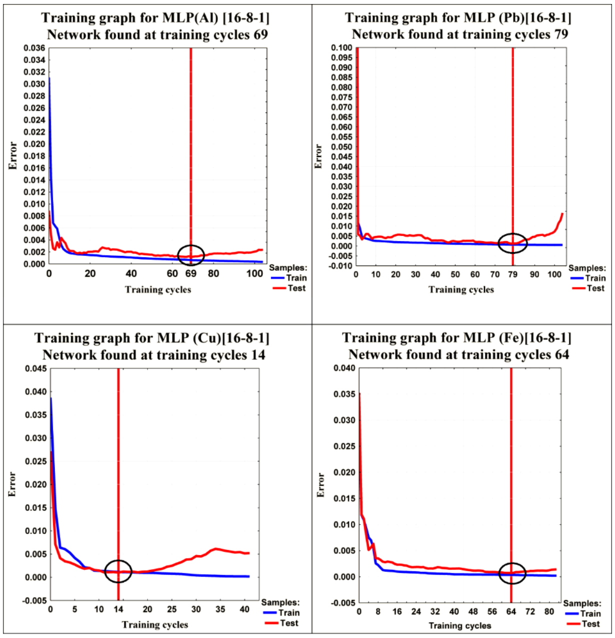

Figure 8. Evolution of the quadratic sum of the errors (function of the cost) with eight neurons in the hidden layer for the four heavy metals

Finally, we can therefore choose eight neurons from the hidden layer of the network to predict its contents. The architecture found is composed of:

Sixteen neurons of the input layer represent the eighteen independent variables eight neurons of the hidden layer.

A single neuron from the output layer represents the contents of each heavy metal studied.

2.4. Evolution of the cost function

The network's training is depicted in Figure 8. It demonstrates that the desired result is achieved after a limited number of iterations (less than 80) (since the BFGS approach is superior in terms of convergence speed). The two curves corresponding to the evolution of the quadratic sum of the errors of the two phases (learning and test) correctly converge towards the quadratic sum of the errors' minimum with eight hidden neurons (SSE).

2.5. Residue studies

The error made by the models established by ANN on an individual in the model construction sample is called the residue.

The study of the residuals ($Y_i - Y_i'$) is fundamental in more than one way: it first makes it possible to identify possibly aberrant observations or observations that play an important role in determining the regression. Then, the study of these residuals is often the only way to empirically verify the validity of the assumptions of the model.

The relationships between the estimated values of heavy metals using neural network models and their residues are shown in Figure 9. The residues acquired by the models established by the method of neural networks are less distributed for the four metals tested, as seen in these figures (close to zero). This shows that the models developed by the recent method, which is based on the principle of artificial neural networks and employs the BFGS algorithm, for predicting the heavy metal contents of surface water in the TAZA zone from physical-chemical parameters are powerful and efficient and that they are widely used in the development of predictive models.

Conclusions

MLP-type artificial neural networks are very powerful tools at the level of prediction for non-linear problems. But they have a major drawback to the choice of network architecture because that choice is often up to the user. In our study, we developed several models based on the qualified high-performance learning algorithm. The results obtained demonstrate that the BFGS algorithm presents the best performances as well as at the level of the statistical indicators at the level of the speed of convergence, as well as the networks of architecture [16-8-1] with functions

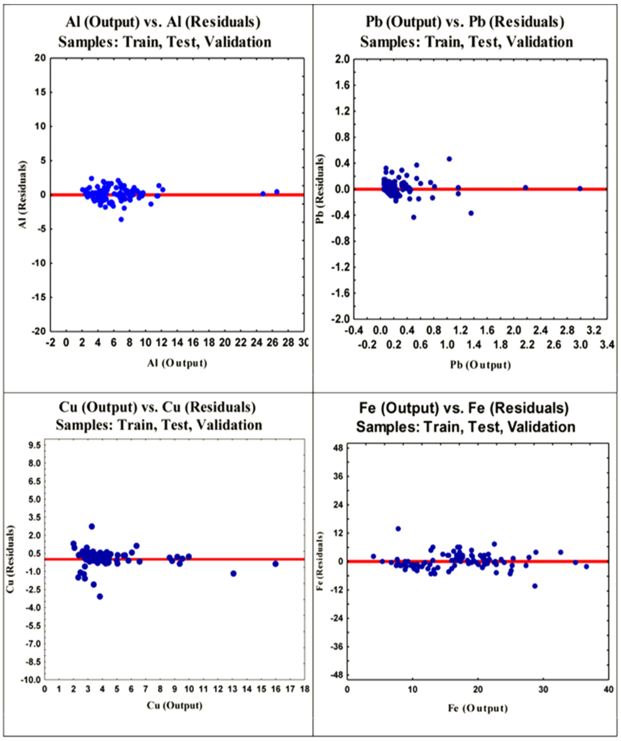

Figure 9. Relationship between the estimated values and their residuals of the RN model relating to the four heavy metals

non-linear activation for the hidden layer of type (logistics for aluminium and iron) and (exponential for Lead and copper). These elaborate neuronal models allowed a very good prediction of heavy metals. This encourages us to consider, in the future, developing other aspects concerning this study.

In perspective, we can continue in the development of this subject through the following suggestions:

- Work on other types of networks, such as recurring networks.
- Use another activation function and another learning algorithm.
- Try to extrapolate the efficient models established by artificial neural networks relating to the prediction of the concentrations of metals studied in surface waters, from the Taza zone to the prediction of the contents of these metals in other Moroccan regions.
- The results obtained prompt us to reflect subsequently on the method, which makes it possible to improve the work accomplished so far. It would be very interesting, for example, to study the possibility of predicting the four parameters at the same time.

References

Abdallaoui, A., & El Badaoui, H. (2015). Comparative study of two stochastic models using the physicochemical characteristics of river sediment to predict the concentration of toxic metals. *Journal of Materials and Environmental Science*, *6*(2), 445–454.

Anagu, I., Ingwersen, J., Utermann, J., & Streck, T. (2009). Estimation of heavy metal sorption in German soils using artificial neural networks. *Geoderma*, *152*(1–2), 104–112. https://doi.org/10.1016/j.geoderma.2009.06.004

Antoine, X., Dreyfuss, P., & Privat, Y. (2007). *Introduction à l'optimisation: aspects théoriques, numériques et algorithmes.* https://math.unice.fr/~dreyfuss/D4.pdf

Asadollahfardi, G., Taklify, A., & Ghanbari, A. (2012). Application of artificial neural network to predict TDS in Talkheh Rud River. *Journal of Irrigation and Drainage Engineering*, *138*(4), 363–370. https://doi.org/10.1061/(ASCE)IR.1943-4774.0000402

Asadollahfardi, G., Zangooi, H., Asadi, M., Jebeli, M. T., Meshkat-Dini, M., & Roohani, N. (2018). Comparison of box-jenkins time series and ANN in predicting total dissolved solid at the Zāyandé-Rūd River, Iran. *Journal of Water Supply: Research and Technology – AQUA*, *67*(7), 673–684. https://doi.org/10.2166/aqua.2018.108

Bayatzadeh Fard, Z., Ghadimi, F., & Fattahi, H. (2017). Use of artificial intelligence techniques to predict distribution of heavy metals in groundwater of Lakan lead-zinc mine in Iran. *Journal of Mining and Environment*, *8*(1), 35–48. https://doi.org/10.22044/jme.2016.592

Ben Abbou, M., El Haji, M., Zemzami, M., & Fadil, F. (2013). Determination de la qualite des eaux souterraines des nappes de la province De Taza (Maroc). *Larhyss Journal*, *16*, 77–90.

Berhail, A. (2016). *Optimisation sans contraintes.* https://dspace.univ-guelma.dz/jspui/bitstream/123456789/640/1/Cours%20Optimisation%20%282017%29-5.pdf

Boudad, B., Sahbi, H., & Boudebbouz, B. (2015). Prédiction de la sécheresse dans le bassin d'Inaouène en utilisant les réseaux de neurones et la régression linéaire multiple. *Journal of Scientific Association for Water Information Systems*, *1*, 13–18.

Boudebbouz, B., Manssouri, I., Mouchtachi, A., & Manssouri, T. (2016). *Utilisation d'un modèle hybride basé sur les réseaux de neurones artificiels-PMC couplés à la décomposition en ondelettes pour la modélisation du régime normale à point de fonctionnement variable. Cas d'une installation industrielle.* HAL Open Science.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, *2*, 303–314. https://doi.org/10.1007/BF02551274

Chamekh, A. (2007). *Optimisation des procédés de mise en forme par les réseaux de neurones artificiels.* https://tel.archives-ouvertes.fr/tel-00445341/document

Coulibaly, P., Anctil, F., & Bobée, B. (1999). Prévision hydrologique par réseaux de neurones artificiels: État de l'art. *Canadian Journal of Civil Engineering*, *26*(3), 293–304. https://doi.org/10.1139/l98-069

Davoudi, E., & Vaferi, B. (2018). Applying artificial neural networks for systematic estimation of degree of fouling in heat exchangers. *Chemical Engineering Research and Design*, *130*, 138–153. https://doi.org/10.1016/j.cherd.2017.12.017

Dreyfus, G. (2006). Les réseaux de neurones. *Revue de l'Electricité et de l'Electronique*, *8*, 37.

El Badaoui, H., Abdallaoui, A., Manssouri, I., & Lancelot, L. (2012). Elaboration de modèles mathématiques stochastiques pour la prédiction des teneurs en métaux lourds des eaux superficielles en utilisant les réseaux de neurones artificiels et la régression linéaire multiple. *Journal of Hydrocarbons Mines and Environmental Research*, *3*(2), 31–36.

El Chaal, R., & Aboutafail, M. O. (2022). A comparative study of back-propagation algorithms: Levenberg-Marquart and BFGS for the formation of multilayer neural networks for estimation of fluoride. *Communications in Mathematical Biology and Neuroscience*, 1–17.

El Haji, M., Boutaleb, S., Laamarti, R., & Laarej, L. (2012). Qualité des eaux de surface et souterraine de la région de Taza (Maroc): bilan et situation des eaux. *Afrique Science: Revue Internationale Des Sciences et Technologie*, *8*(1), 67–78.

El Hmaidi, A., El Badaoui, H., Abdallaoui, A., & El Moumni, B. (2013). Application des réseaux de neurones artificiels de type PMC pour la prédiction des teneurs en carbone organique dans les dépôts du quaternaire terminal de la mer d'alboran. *European Journal of Scientific Research*, *107*(3), 400–413.

Fletcher, R. (1981). Constrained optimization. In *Practical methods of optimization* (Vol. 2). John Wiley & Sons Ltd.

Funahashi, K.-I. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks*, *2*(3), 183–192. https://doi.org/10.1016/0893-6080(89)90003-8

Graupe, D. (2007). *Principles of artificial neural networks* (2^{nd} ed.). World Scientific. https://doi.org/10.1142/6429

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, *4*(2), 251–257. https://doi.org/10.1016/0893-6080(91)90009-T

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8

Ibrahim, M. A. H. Bin, Mamat, M., & June, L. W. (2014). BFGS method: A new search direction. *Sains Malaysiana*, *43*(10), 1591–1597.

Jacobs, R. A. (1988). Increased rates of convergence through learning rate adaptation. *Neural Networks*, *1*(4), 295–307. https://doi.org/10.1016/0893-6080(88)90003-2

Jiang, Y., Zhang, G., Wang, J., & Vaferi, B. (2021). Hydrogen solubility in aromatic/cyclic compounds: Prediction by different machine learning techniques. *International Journal of Hydrogen Energy*, *46*(46), 23591–23602. https://doi.org/10.1016/j.ijhydene.2021.04.148

Karimi, M., Vaferi, B., Hosseini, S. H., Olazar, M., & Rashidi, S. (2021). Smart computing approach for design and scale-up of conical spouted beds with open-sided draft tubes. *Particuology*, *55*, 179–190. https://doi.org/10.1016/j.partic.2020.09.003

Kharroubi, O., Blanpain, O., Masson, E., & Lallahem, S. (2016). Application du réseau des neurones artificiels à la prévision des débits horaires: Cas du bassin versant de l'Eure, France. *Hydrological Sciences Journal*, *61*(3), 541–550. https://doi.org/10.1080/02626667.2014.933225

Mottelet, S. (2003). *RO04/TI07 – Optimisation non-linéaire*. Université de Technologie de Compiègne.

Parizeau, M. (2004). *Reseaux De Neurones GIF-21140 et GIF-64326*. Universite LAVAL.

Patro, S. G. K., & sahu, K. K. (2015). Normalization: A preprocessing stage. *International Advanced Research Journal in Science, Engineering and Technology*, *2*(3), 20–22. https://doi.org/10.17148/IARJSET.2015.2305

Rahnama, E., Bazrafshan, O., & Asadollahfardi, G. (2020). Application of data-driven methods to predict the sodium adsorption rate (SAR) in different climates in Iran. *Arabian Journal of Geosciences*, *13*(21), 1160. https://doi.org/10.1007/s12517-020-06146-4

Shouyu, C., & Honglan, J. (2005). Fuzzy optimization neural network approach for ice forecast in the inner Mongolia reach of the Yellow River. *Hydrological Sciences Journal*, *50*(2), 37–41. https://doi.org/10.1623/hysj.50.2.319.61793

Skansi, S. (2018). *Introduction to deep learning. From logical calculus to artificial intelligence*. Springer International Publishing AG. https://doi.org/10.1007/978-3-319-73004-2

Snyman, J. A., & Wilke, D. N. (2018). *Springer optimization and its applications: Vol. 133. Practical mathematical optimization* (2^{nd} ed.). Springer Cham. https://doi.org/10.1007/978-3-319-77586-9

STATISTICA. (2014, June 4). *Logiciel STATISTICA (version 12.5.192.5)*. http://www.statsoft.fr/logiciels/reseaux-de-neurones-automatises.php

Touzet, C. (2016). *Les reseaux de neurones artificiels, introduction au connexionnisme*. HAL Open Science.

Vaferi, B., Eslamloueyan, R., & Ayatollahi, S. (2011). Automatic recognition of oil reservoir models from well testing data by using multilayer perceptron networks. *Journal of Petroleum Science and Engineering*, *77*(3), 254–262. https://doi.org/10.1016/j.petrol.2011.03.002

Verma, A. K., & Singh, T. N. (2013). Prediction of water quality from simple field parameters. *Environmental Earth Sciences*, *69*(3), 821–829. https://doi.org/10.1007/s12665-012-1967-6

Zhou, Z., Davoudi, E., & Vaferi, B. (2021). Monitoring the effect of surface functionalization on the CO_2 capture by graphene oxide/methyl diethanolamine nanofluids. *Journal of Environmental Chemical Engineering*, *9*(5), 106202. https://doi.org/10.1016/j.jece.2021.106202