Iraqi Journal of Physics, 2025 *Vol.23, No.1, PP. 1-9*
https://doi.org/10.30723/ijp.v23i1.1338 *P-ISSN: 2070-4003*
E-ISSN: 2664-5548

Design an Efficient Neural Network to Determine the Rate of Contamination in the Tigris River in Baghdad City

Farah F. Ghazi[1a*], Luma N. M. Tawfiq[1b], and T. A. Jawad[2c]
[1]*Department of Mathematics, College of Education for Pure Science Ibn Al-Haitham, University of Baghdad. Iraq*
[2]*Department of Mathematics, Kyung Hee University 26 Kyungheedae-ro, Dongdaemun-gu, Seoul 02447, Korea*
[a*]Corresponding author: farah.f.g@ihcoedu.uobaghdad.edu.iq

Abstract
This article proposes a new technique for determining the rate of contamination. First, a generative adversarial neural network (ANN) parallel processing technique is constructed and trained using real and secret images. Then, after the model is stabilized, the real image is passed to the generator. Finally, the generator creates an image that is visually similar to the secret image, thus achieving the same effect as the secret image transmission. Experimental results show that this technique has a good effect on the security of secret information transmission and increases the capacity of information hiding. The metric signal of noise, a structural similarity index measure, was used to determine the success of colour image-hiding techniques within ANN. The results of the ANN were in sequence: 41.2813, 0.6914. The results of the ANN were in sequence 41.2813, 0.6914. These results provide insights into how well the hidden information is concealed within the image and the extent to which the visual integrity of the image is preserved.

Article Info.

Keywords:
Artificial Neural Networks, Physical Attributes, Unconstrained Optimization, Levenberg-Marquardt (LM) Algorithm,Contamination.

Article history:
Received: Nov. 09, 2024
Revised: Jan. 27, 2025
Accepted: Feb. 20, 2025
Published:Mar. 01,2025

1. Introduction
Developing new approaches for coverless information hiding in color images using adversarial neural networks (ANN) trained by deep learning, especially using feed forward neural networks (FFNN) [1, 2], has both societal and research significance. Information in digital form is vulnerable to both adversarial and unauthorized threats. As traditional digital watermarking typically involves embedding the information within traditional media for verification purposes, hiding the information without using any traditional media is significantly beneficial. Although various tools are available for information hiding, including traditional digital watermarking, several challenges while hiding significant information in a digital image [3, 4]. The image-domain methods, which hide information directly into original images, are widely available in real-world applications [4]. Therefore, developing new domain information hiding technology is applicable to the physical world, especially for information-hiding in a color image [5, 6]. Herein the authors exploit deep learning approaches for trained information hiding without using additional media in color images [7]. They pointed out that using this approach reduces the possibility of the step analysis algorithms to identify it. The integration and evolution of generative ANNs have taken this area to the next level, allowing the creation of quasi-transparent step images without the need for a cover image "*Iraqi Ministry of Health and Environment, Results of laboratory tests performed by the Iraqi Ministry of Health and Environment for the Tigris River*" [8]. The Tigris River is one of the seven major sources of potable water in Iraq. Tigris River has a length of about 1718

km. It originates from the Taurus Mountains in eastern Turkey and ends in the Shatt al-Arab. The river length in Iraq is 1,400 km [9-13]. In the Baghdad region specifically, it is about 50 km. It divides the Baghdad region into two parts, Karkh and Rusafa. Recent research has explored the use of neural networks (ANNs) for coverless information hiding in images. For more details, see [12-18].

2. Study Area

The capital of Iraq is Baghdad city (33°14'-33°25'N, 44°31‘-44°17'E), which has a climate of dry, hot in summer and cold in winter; the mean rainfall is about 151.8 mm [19, 20].

The Tigris River water samples of water were collected from seven regions in Baghdad city: Al-Adhamiya (Ś**1**), Al-Shuhada (Ś**2**), Al-Jadriya (Ś**3**), Al-Kuraiyat (Ś**4**), Al-Dora (near the two-storey bridge) (Ś**5**), Karrada Kharj (Ś**6**) and Al-Atifiyah (Ś**7**). The samples were collected from the Tigris River about (20-30) cm below the surface. The samples were put in an ice-cool box to present the laboratory. Metals determinations were done by Inductively coupled Plasma (ICP) [21]. Fig. 1 shows the regions of the collected samples using GIS Arcmap 10.9 program. Table 1 illustrated the results of heavy metals for samples in different regions.

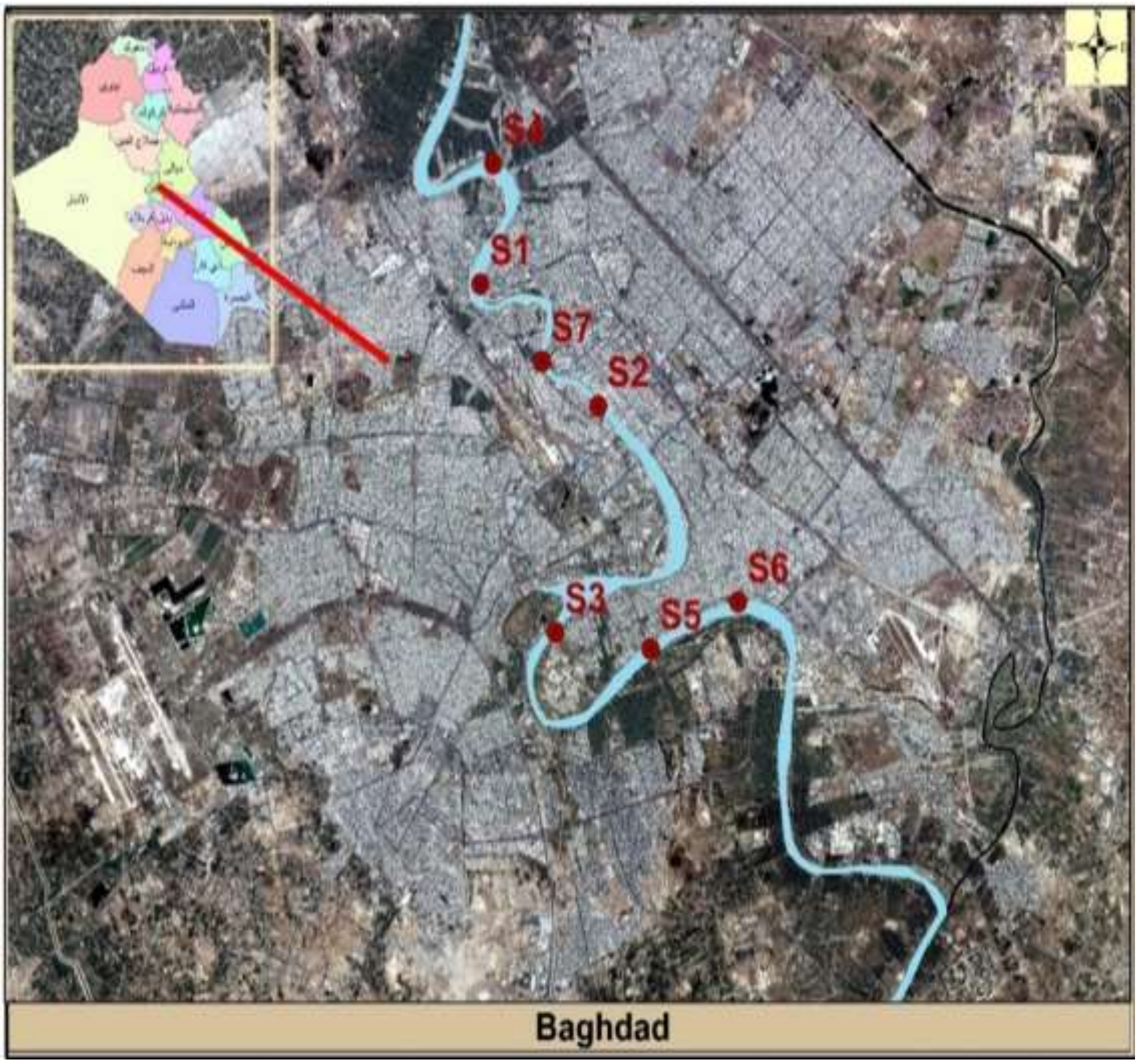

Figure 1: Location of selected samples.

Table 1: The results of heavy metals for samples in different regions.

	Stander rivers for year 25/1967	**Ś1**	**Ś2**	**Ś3**	**Ś4**	**Ś5**	**Ś6**	**Ś7**
PH	6.5-8.5	7.3	7.4	7.3	7.1	7.2	7.4	7.3
TDS (ppm)	-	330	314	305	327	304	300	305
Ec	-	530	490	474	546	505	464	507
TSS (ppm)	-	388	310	200	388	310	94	78
COD (ppm)	-	20	40	20	20	40	20	40
Oil (ppm)	-	11	20.2	14.6	-	-	-	-
NTU (ppm)	-	415	370	250	418	339	405	416
NO3 (ppm)	15	1	1.5	3.1	0.96	1.6	3.36	3.6
Output layer which consisted the continuation of heavy metals in Tigris River in different location of Baghdad city								
Cd (ppm)	0.005	0.048	0.027	0.032	0.01415	0.01271	0.0124	0.01069
Mg (ppm)	0.1	34.54	33.22	31.416	34.095	33.132	32.316	31.787
Pb (ppm)	0.05	0.01078	00.0279	0.01078	0.01078	0	0.0279	0.0285
Fe (ppm)	0.3	0.0222	0.0142	0.0267	0.0222	0.0142	0.0267	0.0195
Cr (ppm)	0.05	0.039	0.024	0.025	0.01597	0.01184	0.01195	0.0124

3. Artificial Neural Networks

A neural network (NN) is a structure of parallel processing networks for distributing information in the form of connected layers consisting of a set of nodes called neurons (also called processing elements), which are the basic processor in ANNs, along with directed line segments between them called links (also are called connections) [22, 23]. All nodes can take any number of arrival connections and can have any number of coming-out connections, but the signs must be the same [24]. All nodes have one coming-out connection that can branch out to form multiple output connections, each carrying the same sign. Each node possesses a transfer (activation) function that can use input signs and produces the node's output sign. Generally, ANN is a generalization of mathematical models of the human brain based on processing information that occurs at many connection nodes; signs are passed between nodes over connection links which have an associated weight; each node applies a transfer function to its weighted input net to determine its sign of output [25].

The treatment of given data hold by income these data as weighted input vector x, as form $W_j^T x$ to enter in hidden layers. It is assumed that each hidden neuron has the same activation function σ, but that bias bj. So the output of j^{th} hidden neuron in the hidden layer is $\sigma(W_j^T x + b_j)$, and again weighted by-product with ʋj entering the output layer in the form:

$$g(x) = \sum_{j=1}^{k} \upsilon j \ \sigma\left(W_j^T x + b_j\right) \quad (1)$$

where g(x) represents the output of the NN. Note σ must be choosing sigmoidal functions, so herein suitable sigmoidal σ was chosen, which is defined as [26]:

$$\sigma(n_i) = \frac{2}{e^{-2ni} + 1} - 1 \quad (2)$$

Therefore, the ANN input-output equation is: $\hat{Y} = \Phi(x^T W^T + b^T)\,\upsilon^T$. where $W \epsilon R^{n\times r}$ is adjustable input weights, $\upsilon \epsilon R^{1\times n}$ is adjustable output weights, and $b \epsilon R^{n\times r}$ is biased.

The structure of ANN interconnections can be classified into different types of ANN architecture, such as feed-forward neural network (FFNN). Nodes are organized in the form of layers; each receives input from the previous layer, and its output is fed to the next layer. The data goes from the input node to the output node in feed-forward way, i.e., forward loops. In a Feedback Neural Network (FBNN), all possible connections between layers and neurons are allowed. The data is transferred to the network as back loops. FFNN was chosen for this work, as illustrated in Fig. 2.

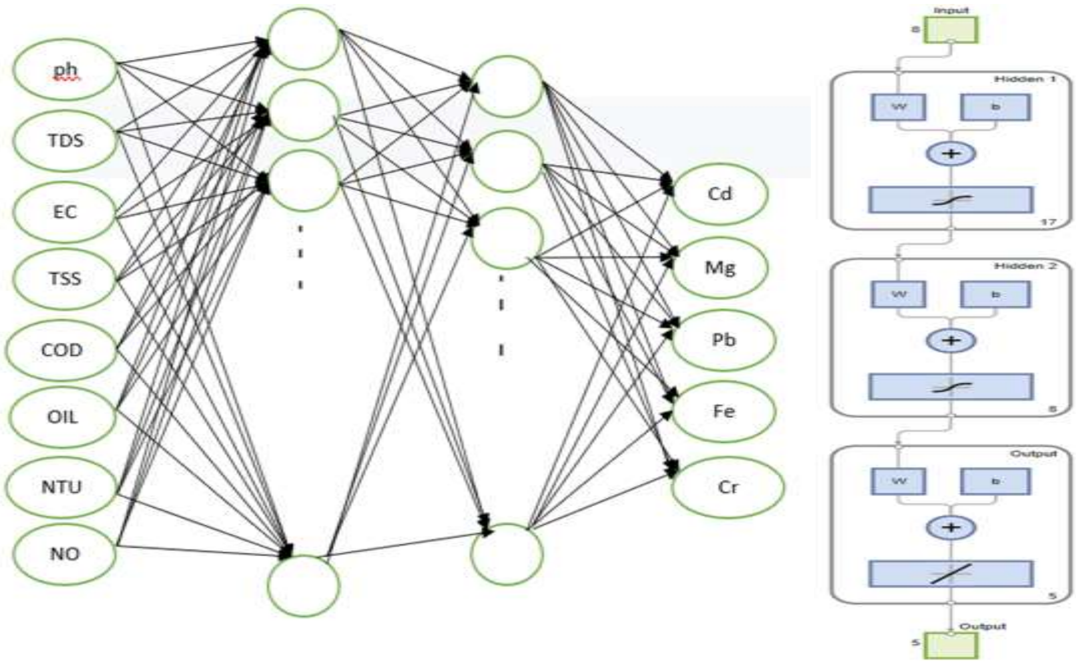

Figure 2: Architecture of proposed design.

4. Application

The used FFNN included three fully interconnected layers: input, hidden and output. The input layer consists of 5 neurons to determine the contamination rate in two cases: the training case (periodic) and the testing case (non-periodic). However, in the hidden layer, different numbers of layers were taken, such as in the training case (periodic), which takes (2, 3 and 4) layers, each consisting of 10 neurons. While the testing case (non-periodic) takes 2 layers in two cases, the first [10 5] neurons and the second [10 10] neurons (depending on trial and error) with tanhsig. activation function (Eq. (2)), and the output layer consists of 1 neuron with linsig. activation function (see Fig.1). Train the suggested ANN using the back-propagation rule and choose a new MLM algorithm.

5. LM Training Algorithm

Levenberg-Marquardt (LM) method is an approach to training with 2nd order speed without need to compute the Hessian matrix. When the error function (performance function) has the form of a sum of squares, then the approximate of Hessian matrix can be define as $= J^T J$, where J is the Jacobian matrix, which contains first derivatives of the performance function with respect to the weights, and the gradient can be calculated as g= $J^T E(w_k)$, where $E(w_k)$, is a vector of errors of network.

The LM method employs the following approximation to the Hessian matrix [27]:

$$H \approx J^T J + \mu I \quad (4)$$

where μ is the combination coefficient and it is always positive.

The updates form of the weight by LM is [28]:

$$w_{k+1} = w_k - (J^T J + \mu I)^{-1} J_k^T e \quad (5)$$

To motivate this, LM algorithm can be interpreted as approximating $E(x+p)$ by a quadratic form using Taylor expansion:

$$E(x+p) \approx E(x) + p^T g(x) + \frac{1}{2} p^T (J^T J + \mu I) p \quad (6)$$

An alternative view is to minimize the quadratic as a function of p. If $J^T J + \mu I$ is positive definite, then the minimum is obtained by setting the derivative equal to zero. If $J^T J + \mu I$ is indefinite, then the quadratic function does not have a finite minimum.

Herein a simplified algorithm for the training process by the Levenberg-Marquardt (LM) rule defined in MATLAB built as "trainlm":

1. Initialization Process:
   - Initial choose of the weights (W) and biases (b) are randomly
   - Take the learning rate ($\eta = 0.01$) for the LM algorithm.
2. Forward Propagation:
   - For each input sample x_i:
   - Calculate the weighted sum then apply the transfer function for each node (neuron) in the hidden layer: $a_{ij} = \sum_{k=1}^{n} w_{ijk} x_{ik} + b_{ij}$

$$u_{ij} = \sigma(a_{ij})$$

   - Propagate the activations to the output layer using a similar process:
   - $a_{ik} = \sum_{j=1}^{m} w_{ijk} a_{ij} + b_{ik}$

$$u_{ik} = \sigma(a_{ik})$$

3. Error Calculation:

The error (E_i) must be calculated between target output ($\dot{u}_{ik}$) and predicted output (u_{ik})

$$E_i = \frac{1}{2} \sum_{k=1}^{K} (\dot{u}_{ik} - u_{ik})^2$$

4. Back-propagation Process:
   - The gradient of the error with respect to weights and biases must be calculated for the output layer as: $g_{ik} = -(\dot{u}_{ik} - u_{ik})\, \acute{\sigma}(a_{ik})$

$$\frac{\partial E_i}{\partial w_{ijk}} = g_{ik} a_{ij} \quad ; \quad \frac{\partial E_i}{\partial b_{ik}} = g_{ik}$$

   - Then propagate the gradient of error back to the hidden layer and again calculating the gradients there

$$g_{ij} = \acute{\sigma}(a_{ij}) \sum_{k=1}^{K} w_{ijk} g_{ik}$$

$$\frac{\partial E_i}{\partial w_{ijk}} = g_{ij} x_{ik} \quad ; \quad \frac{\partial E_i}{\partial b_{ij}} = g_{ij}$$

5. Update weights and biases using Levenberg-Marquardt (LM) rule:
   - The weights and biases are updated using the LM update rule:

$$w_{ijk}^{(t+1)} = w_{ijk}^{(t)} - (J^T J + \lambda I)^{-1} J_k^T e$$

$$b_{ijk}^{(t+1)} = b_{ijk}^{(t)} - \eta \frac{\frac{\partial E_i}{\partial b_{ij}}}{\sum_{k=1}^{K} \frac{\partial E_i}{\partial b_{ij}}}$$

6. Repeat:

- Iterate the process through the dataset for multiple times, to adjusting the weights and biases after the beginning in each iteration.
- Stop the train process when the error becomes fewer or a predefined number of iterations is reached.

Depending on its speed, the suggested algorithm seems to be the most efficient way to training FFNNs of moderate size (with up to several hundred weights). Additionally, it has a streamlined implementation in MATLAB software, where the solution presented and built-in as the matrix equation. These attributes make it particularly effective in a MATLAB environment.

7. Results and discussion

The performance and accuracy of the solution for suggested ANN is measured by the computing the error in training, validation and testing case which equal to perf = 2.466568e-05.

To check the efficiency of the neural solutions in these three cases, the mean square error (mse) was calculated for different values of the epochs, presented in Fig. 3. The fineness of the training process with variation value of epoch and time is illustrated in Table 2. Moreover, the target of output in trained, tested and validation cases is shown in Fig. 4. Also, Fig. 5 illustrate the manner of gradient in validation case at epoch 100000 which stopped in epoch 14 because it reached to minimum values.

Finally, the laboratory value for the rate of heavy metals in Tigris River is of high accuracy compared with the value of the output for the suggested neural network, as shown in Fig. 6.

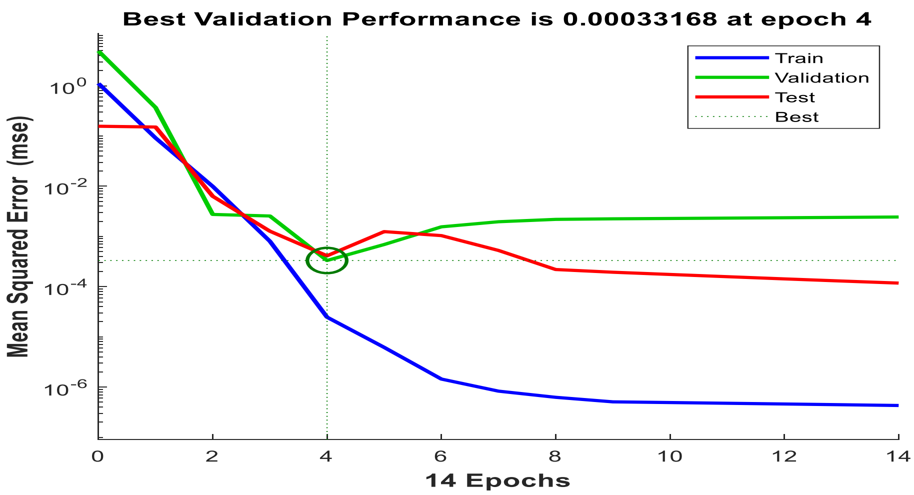

Figure 3: MSE in train, test and validation casas.

Table 2: MSE for training, validation and testing cases.

Type	Target values	MSE
Training	65	2.466568e-05
Validation	15	3.31682267e-04
Testing	20	4.118766753e-04

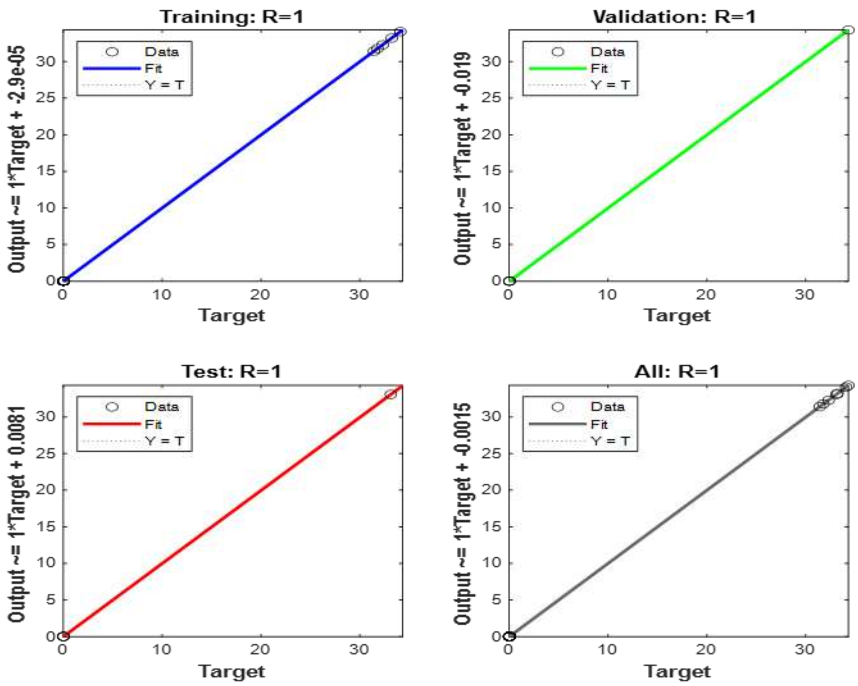

Figure 4: Target of the results in train, test and validation case.

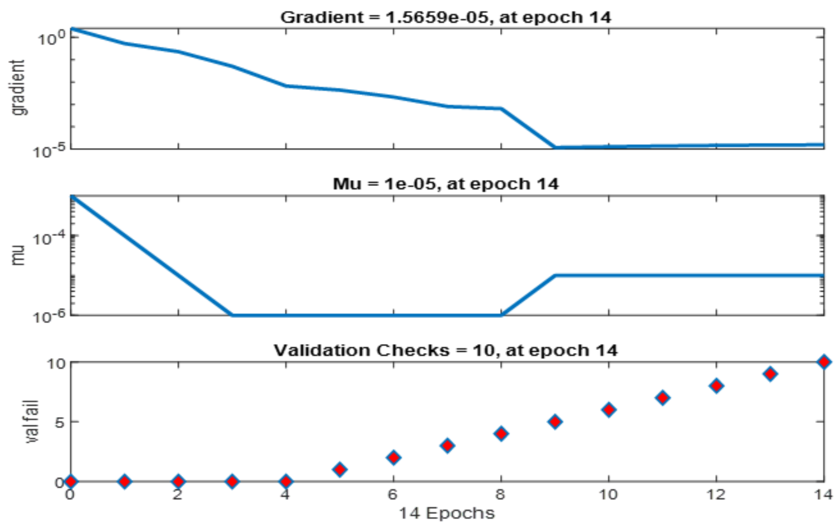

Figure 5: Behaviour of GD at epoch 100000 in validation case.

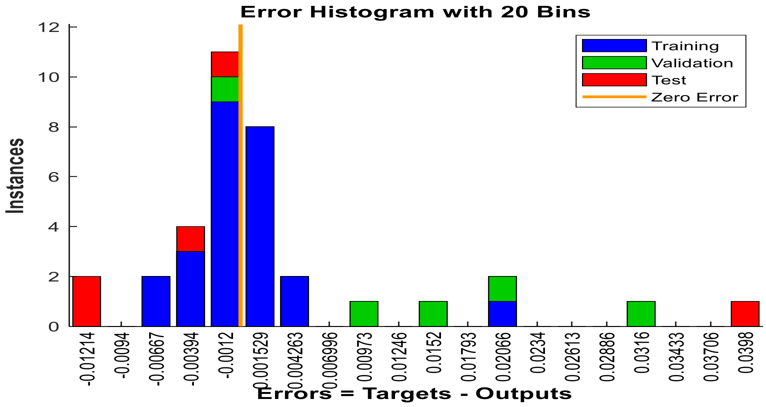

Figure 6: The error between laboratory values and the neural network values.

7. Conclusions

A neural network was used to consider the contamination of the Tigris River in Baghdad city, using the chemical and physical properties of the river as input data, and the output represented the concentrations of the heavy metals in the river. The error between laboratory values and the neural network values is small, less than 0.0012. So, ANN can be a useful tool for estimating the rate of heavy metals in the river without laboratory calibration.

Conflict of Interest

The authors declare that they have no conflict of interest.

References

1. F. F. Ghazi, Iraqi J. Sci. **61**, 2663 (2020). DOI: 10.24996/ijs.2020.61.10.23.
2. L. N. M. Tawfiq and M. A. Hassan, J. Phys.: Conf. Ser. **1003**, 012057 (2018). DOI: 10.1088/1742-6596/1003/1/012057.
3. R. R. Al-Ani, A. Al Obaidy, and R. M. Badri, Int. J. Adv. Res. **2**, 1125 (2014).
4. A. N. Ismael, H. A. Abed, and M. A. Abed, Adv. Comp. Sig. Sys. **4**, 25. DOI: 10.23977/acss.2020.040105.
5. M. A. Ibrahim, M. J. Mohammed-Ridha, H. A. Hussein, and A. a. H. Faisal, Iraqi J. Agricult. Sci. **51**, 1572 (2020). DOI: 10.36103/ijas.v51i6.1184.
6. L. N. Tawfiq and F. F. Ghazi, Int. J. Innovat. Sci. Eng. **11**, 58 (2015).
7. F. F. Ghazi, J. Phys. Conf. Ser. **1003**, 012058 (2018). DOI: 10.1088/1742-6596/1003/1/012058.
8. L. Naji Mohammed Tawfiq, N. H. Al-Noor, and T. H. Al-Noor, J. Phys.: Conf. Ser. **1294**, 032020 (2019). DOI: 10.1088/1742-6596/1294/3/032020.
9. L. Tawfiq, K. A. Jasim, and E. Abdulhmeed, Glob. J. Eng. Sci. Res. **3**, 75 (2016).
10. F. F. Ghazi, R. I. Khalaf, and N. Sabah, Al-Qadisiyah J. Pure Sci. **25**, 60 (2020). DOI: 10.29350/ jops.2020.25. 2.1107
11. L. N. Tawfiq, Y. a. J. I. a.-H. J. F. P. Oraibi, and A. Science, **26**, 275 (2017).
12. F. F. Ghazi and L. N. M. Tawfiq, Int. J. Optimiz. Cont. Theor. Applicat. (IJOCTA) **14**, 249 (2024). DOI: 10.11121/ijocta.1519.
13. N. a. K. Hussein and B. Al-Sarray, Iraqi J. Sci. **63**, 3153 (2022). DOI: 10.24996/ijs.2022.63.7.36.
14. L. N. Tawfiq and M. M. Hilal, Ibn AL-Haitham J. Pure Appl. Sci. **27**, 501 (2017).
15. F. Regazzoni, L. Dedè, and A. Quarteroni, J. Computat. Phys. **397**, 108852 (2019). DOI: 10.1016/j.jcp.2019.07.050.
16. K. Blekas and I. E. Lagaris, SN Appl. Sci. **2**, 1954 (2020). DOI: 10.1007/s42452-020-03713-y.
17. M. Pakdaman, A. Ahmadian, S. Effati, S. Salahshour, and D. Baleanu, Appl. Math. Comput. **293**, 81 (2017). DOI: 10.1016/j.amc.2016.07.021.
18. T. T. Dufera, Mach. Lear. Appl. **5**, 100058 (2021). DOI: 10.1016/j.mlwa.2021.100058.

19. E. Jamili and V. Dua, Comp. Chem. Eng. **147**, 107221 (2021). DOI: 10.1016/j.compchemeng.2020.107221.
20. T. Knoke and T. Wick, Examp. Counterexam. **1**, 100035 (2021). DOI: 10.1016/j.exco.2021.100035.
21. L. N. Mohammed Tawfiq and M. A. Hasan, Ibn AL-Haitham J. Pure Appl. Sci. **32**, 132 (2019). DOI: 10.30526/32.1.1979.
22. R. Gubta, E. Gubta, and C. B. Batra, Baghdad Sci. J. **19**, 0865 (2022). DOI: 10.21123/bsj.2022.19.4.0865.
23. Z. K. Hussien and B. N. Dhannoon, Baghdad Sci. J. **17**, 0701 (2020). DOI: 10.21123/bsj.2020.17.2(SI).0701.
24. R. S. Muhammad and M. I. Younis, Iraqi J. Sci. **58**, 355 (2022).
25. L. N. M. Tawfiq and M. H. Ali, J. Phys.: Conf. Ser. **1530**, 012067 (2020). DOI: 10.1088/1742-6596/1530/1/012067.
26. D. A. Anggoro and S. Hajiati, Iraqi J. Sci. **64**, 907 (2023). DOI: 10.24996/ijs.2023.64.2.35.
27. R. Naoum, L. N. M., Ibn AL-Haitham J. Pure Appl. Sci. **20**, 146 (2017).
28. S. S. Mahmood, J. H. Eidi, and J. A. Jasem, Baghdad Sci. J. **19**, 0297 (2022). DOI: 10.21123/bsj.2022.19.2.0297.

تصميم شبكات عصبية كفوءة لتحديد نسبة التلوث في نهر دجلة في مدينة بغداد

فرح فيصل غازي[1] و لمى ناجي محمد توفيق[1] وثابت عبد الجواد[2]
[1]قسم الرياضيات، كلية التربية للعلوم الصرفة ابن الهيثم، جامعة بغداد، العراق
[2]قسم الرياضيات، جامعة *Kyung Hee*، سيؤول، كوريا الجنوبيه

الخلاصة

اقترحنا تصميم شبكة عصبية صناعية على اساس خوارزميات تعلم جديدة. ثم استخدمت لحساب نسبة تلوث في نهر دجلة في مدينة بغداد. معمارية التصميم تحتوي 4 طبقات: طبقة الادخال تحتوي 8 عقد مدخلات تمثل بيانات حول PH, مجموع ذوبان المواد الصلبة (T.D.S)، احتياج الاوكسجين الكيميائي (COD)، الاتصال الكهربائي (EC)، مجموع المواد الصلبة العالقة (T.S.S)، نفط، وحدة قياس العكرة (NTU)، نترات (NO) الطبقة الخفية الاولى تتضمن 17 عقدة، الطبقة الخفية الثانية تتضمن 8 عقد مع دالة الانتقال .tansig لكل طبقة خفية و طبقة الاخراج ذات دالة الانتقال الخطية تتضمن 5 عقد تمثل بيانات حول نسب الكادميومCd، المغنيزيوم Mg، رصاص pb، حديد Fe و الكروم Cr. التصميم المقترح يمتلك ميزات منها: النتائج تتقارب بشكل سريع لدالة الطاقة والتي تكون دوال محدبة ايضا استخدمنا طريقة البحث الخطي الغير رتيبة للحصول على طول الخطوة المثالي في معالجة التعليم (نسبة التعلم). اداء الشبكة المقترحة قورنت مع الفحوصات المختبرية التقليدية باستخدام عينة البيانات (عينات التدريب و الاختبار). النتائج لهذا العمل اثبتت الشبكات المقترحة دربت على مقاييس تجريبية طبقت بنجاح لتخمين دقيق و سريع لنسبة التلوث في نهر دجلة.

الكلمات المفتاحية: شبكات عصبية صناعية، خواص فيزيائية، امثلية غير مقيدة، خوارزمية Levenberg- Marquardt (LM)، التلوث.