How to select superior Neural Network simulating inner city contaminant transport? Verification and validation techniques.

A. Wawrzynczak[1,2][0000−0001−8292−6875] and
M. Berendt-Marchel[1][0000−0001−7204−7367]

[1] Institute of Computer Sciences
Siedlce University, Poland awawrzynczak@uph.edu.pl
[2] National Centre for Nuclear Research
Swierk-Otwock, Poland

Abstract. Artificial neural networks (ANNs) can learn via experience to solve almost every problem. However, the ANN application in a new task entails a necessity to perform some additional adaptations. First is fitting the ANNs type or structure by applying the various number of hidden layers and neurons in it or using different activation functions or other parameters, allowing ANN to learn the stated task. The second is the validation and verification methods of the ANN quality that should be suited to the stated task. Occasionally the differences between the ANNs output are significant, and it is easy to choose the best network. However, sometimes the differences pronounced by standard performance parameters are minor, and it is difficult to distinguish which ANN has reached the best level of training. This paper presents the results of training the ANN to predict the spatial and temporal evolution of the airborne contaminant over a city domain. Statistical performance measures have validated the trained ANNs performance. Finally, new measures allowing to judge of both time and spatial distribution of the ANN output have been proposed and used to select the prior ANN.

Keywords: Neural network model · Validation methods · Dispersion model .

1 Introduction

The study presented in this paper was initiated by the willingness to create an emergency-response system able to localize the airborne toxin source in the urban terrain in real-time. The most probable contamination source location should be indicated based on the concentration data reported by the sensor network. Moreover, the process should be quick to ensure the fast action of the emergency response group. In the literature, the process of the contamination source localization based on the outcome is classified as the backward problem and referred to as source term estimation (STE), e.g., [1]. The goal is to find the best or most

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_35

likely match between the predicted (by the applied dispersion model) and observed data, i.e., concentration in the sensor location. Consequently, the model parameters space scanning algorithm guided by the likelihood function is used. This requires many thousands of dispersion model runs. In [2] the localization of the contaminant in the highly urbanized terrain using the approximate Bayesian computation algorithm is presented. Although the results are satisfactory, the computational time of the reconstruction is extended. Thus, urban reconstruction in real-time is not possible, even with the distributed system.

The solution might be an application of the trained Artificial Neural Networks (ANNs) in the place of the dispersion model in the STE algorithm. ANNs are learning by example. Thus, they can be skilled with known examples to solve almost any task. Once well-trained ANNs can solve the stated task very quickly. These characteristics make the ANNs an excellent tool in real-time working systems. The ANN must learn to simulate airborne contaminant transport to be used in the emergency response localization system. The contaminant concentration distribution function is multidimensional and depends on spatial coordinates and time. Additionally, its value depends on external parameters like the contaminant source characteristics (location, release rate, release duration), meteorological conditions, and the domain's geometry. The challenge is the urban geometry, which is very complicated as far the wind field structure on which the contaminant is spread is site-dependent. The ANN training is computationally expensive, but once trained, the ANN would be a high-speed tool that estimates the contaminant concentration distribution.

The first results confirming that ANN has the potential to replace the dispersion model in the contaminant source localization systems are presented in [3]. The comparison of various architectures of ANNs in forecasting the contaminant strength correctly is presented in [4]. The results revealed that standard performance measures like correlation R and mean square error are fallible in pointing out the quality of ANN. In none of the mentioned papers the more profound validation of the proposed ANN models was not presented. This work is aimed at fulfilling at least some of these gaps.

Thus, apart from the known statistical measures, we propose the new ones being able to verify the dynamic agreement between the ANN output and the target both in space and time. Moreover, we propose a method to improve the trained ANN quality when a value of zero represents a large proportion of the training data.

2 ANN model

Feedforward neural networks (FFNNs) are often applied for prediction and function approximation. The first layer of FFNNs consists of the neurons representing the input variables based on which the network should produce the neurons in the output layer. Between the input and output layers, the hidden layers are placed. ANN performance depends on the chosen architecture, i.e., the number of neurons, hidden layers, and the structure of connections. The aim is to

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_35

teach the ANN to predict the contaminant concentration at a specific time and location for the assumed release scenario. Thus, the structure of the input vector is following $Input_i \equiv \{X_s, Y_s, Q, d, x, y, v, t\}$. Based on the input vector for the contamination source at the coordinates (X_s, Y_s) (in meters within a domain) and release lasting through d seconds with the release rate equal Q under wind blowing from the v direction the trained network should return the output neuron $Output_i \equiv C_i^{S_j(x,y)}(t)$ denoting the concentration C at sensor S_j with coordinates (x, y) in t -seconds after starting the release.

3 Domain, Training, and Testing Dataset. Data Preprocessing

The central part of London was chosen as a domain for the training dataset generation. The reason was a willingness to train the ANN using the real field tracer experiment DAPPLE. Unfortunately, about 600 point concentrations were insufficient to train the ANN properly. Thus, the learning dataset was generated using the QUIC Dispersion Modeling System [5]. The details on the domain and simulations setup are presented in [4] with the difference that the assumed release rate is within interval $Q \in \langle 100mg, 999mg \rangle$.

The obtained set covering about 5×10^7 vectors was divided into training - 66%, validation and testing datasets 17% each. The target function is a multidimensional and time-dependent function. However, the neurons in the input layer do straightforwardly reflect this time dependency. Each input vector corresponds to the concentration for a fixed point in time and space for a unique release scenario. The data included in the training and validation dataset were randomly drawn from the whole dataset. However, the testing dataset was carefully selected to judge how well the trained ANN reflects the time dynamics, i.e., whether the ANNs prediction is correct in subsequent time intervals. Thus, the testing dataset contains the vectors covering the whole 67 simulations. To appropriately validate and compare the ANNs, the same testing dataset was used to estimate the performance measures described in Section 4 for all analyzed ANNs architectures.

To give all variables equal weight in the input neuron vector, they have been scaled to the interval $\langle 0, 1 \rangle$. In addition, the target concentration was logarithmized [6]. Moreover, the noise was introduced to the target concentration C and release rate Q. These two variables were chosen for noise introduction due to their inseparable connection. The spatial distribution of concentration on the sensors will depend on the strength of the released substance. The noise was introduced after normalizing the ANN input data as $\dot{C} = C \pm \delta \times C$, where δ was drawn uniformly from the interval $\langle 0, 15 \rangle$.

4 ANN model validation

The model validation aims to evaluate how useful a model is for a given purpose, thereby increasing confidence in model outputs. The verification and validation

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_35

Table 1: The measures calculated for the ANNs models. The ANN colors denote the best values of the measures among other ANNs for training using the noised and orginal data.

Network	trained using noised data				trained using orginal data			
	24-16-8	24-16-8-4-2	24-20-16-12-8-4-2	48-24-16-8-4	24-16-8	24-16-8-4-2	24-20-16-12-8-4-2	48-24-16-8-4
R training	0.8296	0.8381	0.8591	0.8787	0.8780	0.8814	0.9008	0.9292
R test	0.5881	0.7922	0.7054	0.5549	0.7517	0.7613	0.6765	0.4695
MSE	0.0252	0.0242	0.0213	0.0185	0.0189	0.0184	0.0156	0.0111
RMSE×10^{-7}	28.21	3.25	3.26	3.31	3.25	3.25	4.11	10.79
CE	-73.729	0.0065	-0.0006	-0.0303	0.0071	0.0066	-0.5834	-9.9266
$\rho(d^{1:t}_{ANN}, d^{1:t}_{target}))$	0.2027	0.1533	0.1587	0.1795	0.7029	0.6639	0.9616	0.9689
$\overline{MSSDLE_{\vec{x}}}$	0.1508	0.1016	0.1008	0.1251	0.6417	0.6031	0.8924	0.8920
$\overline{MSSDLE_{\vec{y}}}$	0.1460	0.0963	0.0994	0.1211	0.6515	0.5871	0.8886	0.8882
$\overline{MSDLE}$	0.1718	0.1157	0.1181	0.1399	0.6663	0.6611	0.8790	0.8790

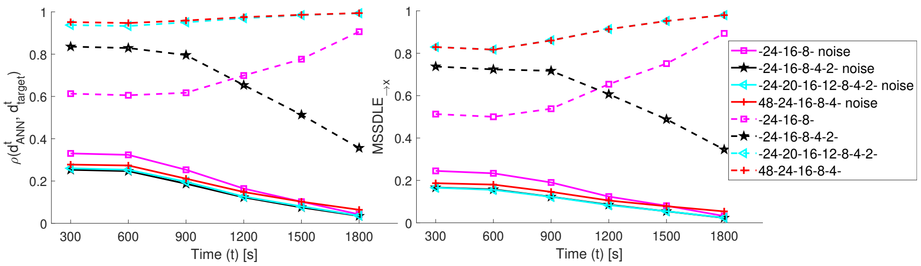


Fig. 1: The measure (a) $\rho(d^t_{ANN}, d^t_{target})$ (Eq. 1), and (b) $MSSDLE_{\vec{x}}$ (Eq. 3)in subsequent time steps for a few considered ANNs. Profile of the measure $MSSDLE_{\vec{y}}$ is analogous. The solid lines correspond to the networks trained using noised data, while the dashed lines correspond to the ANNs trained using original data.

process is indispensable when the single best model has to be indicated from the subset of models. The selection is more difficult when the primary differences are minor. Here we propose the measures helpful to validate which of the trained ANNs has learned to predict best the spatial and time evolution of the target function.

The most common measure used to judge the level of ANN training is the correlation coefficient R between the actual output and the output predicted by the ANN. Usually, after training different ANNs, the final one is chosen based on the highest value of R. The $R = 1$ denotes the ideal fit. The twin measure is the mean square error $MSE = \frac{1}{n}\sum_{i=1}^{n}(C_i - \hat{C}_i)^2$ denoting difference between the ANN output $\hat{C}_i$ and target C_i. The MSE is typically used as the stopping criterion for the ANN training process. However, using MSE means assuming that the underlying data has been generated from a normal distribution. In reality, a dataset rarely fulfills that requirement. It is better to report $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(C_i - \hat{C}_i)^2}$, rather MSE; because RMSE is measured in the same units

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_35

as the original data, and is thus more representative of the size of a 'typical' error. To distinguish the ANNs quality deeper, more measures should be considered.

Model Selection Measure The coefficient of efficiency (CE) is one of the model selection measures: $CE = 1 - \frac{\sum_{i=1}^{n}(C_i - \hat{C}_i)^2}{\sum_{i=1}^{n}(C_i - \bar{C})^2}$. CE is intended to range from zero to one, but negative scores are also permitted. The maximum positive score of one represents a perfect model. The negative scores are unbounded, indicating that the model performs worse than a 'no knowledge' model. CE is sensitive to differences in the observed and modeled means and variances.

Measures estimating the time dynamic The statistical measures described above do not provide information on the quality of the spatial and temporal distribution of the ANN prediction. Therefore, we propose introducing additional measures to verify whether trained ANN can correctly reproduce the concentration gradient (both spatial and time). The focus is put on the agreement in the successive intervals of simulations. The first proposed formula based on a fractional bias is:

$$\rho(d_{ANN}^{1:t}, d_{target}^{1:t}) = \frac{1}{SN} \sum_{j=1}^{SN} \left[\frac{1}{t} \sum_{i=1}^{t} \frac{|C_i^{Sj} - \hat{C}_i^{Sj}|}{C_i^{Sj} + \hat{C}_i^{Sj}} \right], \tag{1}$$

with assumption that if $C_i^{Sj} = 0$ and $\hat{C}_i^{Sj} = 0$ then fraction $\frac{|C_i^{Sj} - \hat{C}_i^{Sj}|}{C_i^{Sj} + \hat{C}_i^{Sj}} = 0$. In Eq. 1 i denotes the subsequent time intervals in which the concentration in Sj point representing the sensor location is estimated. The SN indicates the total number of sensors, $\hat{C}_i^{Sj}$ concentration in time i in point Sj of domain predicted by ANN, while C_i^{Sj} the represents the target concentration. The measure ρ fits into the interval $[0, 1]$. If the ANN model prediction is ideal, then $\rho = 0$, and if the model predictions are completely wrong, it equals 1.

The following measure Mean Squared Derivative Logarithmic Error (MSDLE) is proposed to describe the level of agreement of the target function change in the time between modeled dataset and observed in each point of the 2D space. In each point Sl of the space the $MSDLE$ is calculated as follows:

$$MSDLE(Sl) = \frac{1}{M} \sum_{m=1}^{M} \left[\frac{1}{t-1} \sum_{j=2}^{t} \left(\frac{\ln(C_j^{Sl,m} - C_{j-1}^{Sl,m}) - \ln(\hat{C}_j^{Sl,m} - \hat{C}_{j-1}^{Sl,m})}{\ln(C_j^{Sl,m} - C_{j-1}^{Sl,m}) + \ln(\hat{C}_j^{Sl,m} - \hat{C}_{j-1}^{Sl,m})} \right)^2 \right]. \tag{2}$$

The M denotes the number of simulations run over t time steps, $\hat{C}_j^{Sl,m}$ the ANN prediction of the target value in time j at the point Sl for the simulation m. The observed value is $C_j^{Sl,m}$. The result of Eq. 2 is the 2D map of the measure distribution. To characterize the measure by a single value, the averaging over

DOI: 10.1007/978-3-031-36021-3_35

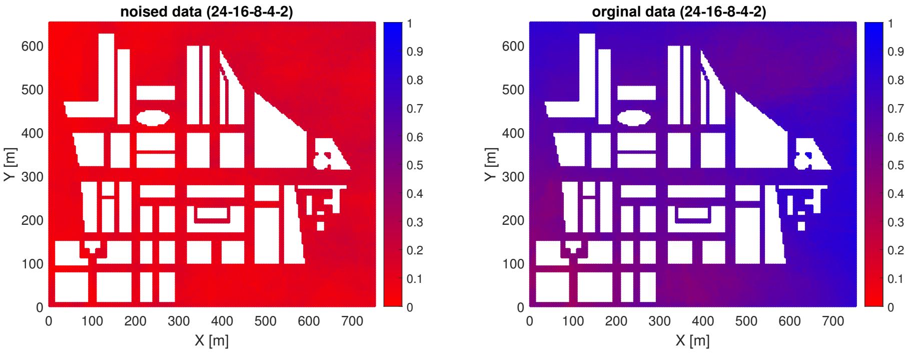
the number of SN space points is performed $\overline{MSDLE} = \frac{1}{SN}\sum_{l=1}^{SN} MSDLE(l)$. The $\overline{MSDLE}$ is scaled to the $[0,1]$ interval and equals 0 for the ideal model. The last proposed measure is aimed to represent how well in each time-step t the spatial gradient of the target function is reproduced. The Mean Squared Spatial Derivative Logarithmic Error (MSSDLE) is estimated in two main directions and $\vec{y}$ of the 2D domain. The corresponding formula for $\vec{x}$-direction is following:

$$MSSDLE_{\vec{x}}(t=j) = \frac{1}{M}\sum_{m=1}^{M}\Bigg[\frac{1}{N_L}\sum_{L=1}^{N_L}\Bigg[\frac{1}{N_K-1} \sum_{K=2}^{N_K}\left(\frac{\ln\left(\frac{(C_j^{m(K,L)}-C_j^{m(K-1,L)})}{\Delta K}\right)-\ln\left(\frac{(\hat{C}_j^{m(K,L)}-\hat{C}_j^{m(K-1,L)})}{\Delta K}\right)}{\ln\left(\frac{(C_j^{m(K,L)}-C_j^{m(K-1,L)})}{\Delta K}\right)+\ln\left(\frac{(\hat{C}_j^{m(K,L)}-\hat{C}_j^{m(K-1,L)})}{\Delta K}\right)}\right)^2\Bigg]\Bigg]. \quad (3)$$

The N_K denotes the number of points on a grid in $\vec{x}$ direction. Taking into account the presence of buildings in the domain, the distance between the points of the grid (sensors) $\Delta K\left((x_1,y_1),(x_2,y_2)\right) = |x_2-x_1|$ is included in the measure. The $MSSDLE_{\vec{y}}(t=j)$ is calculated analogously in the $\vec{y}$ direction. To represent the measure by a single value, the averaging over the time steps of simulations T is performed $\overline{MSSDLE} = \frac{1}{T}\sum_{t=1}^{T} MSSDLE(t)$. The $\overline{MSSDLE}$ is scaled to the $[0,1]$ interval and equals 0 for the ideal model.

5 Results

We have trained the multiple ANNs using the dataset described in Section 3 and Matlab Deep Learning Toolbox. Among tested activation functions in the hidden layers, the *hyperbolic tangent*, and the linear function in the output layer performed the best. The network training was stopped at the lowest possible MSE of the validation test, assuming the upper limit of epochs to 70, with the target MSE set to $1e-08$ value. We have trained the ANNs with the same architectures using the original and noised datasets. The measures described in Section 4 were calculated for each developed ANN model. Table 1 presents the values of the estimated measures for four ANNs with the highest R-value. The differences in R-values are pretty slight. In such cases, selecting the prior ANN is complicated and must be done carefully. The proposed additional measures should help facilitate the selection of the best-trained ANNs. Table 1 is divided into two parts. The left side presents the measures for the ANNs trained using the noised data, and the right side for the same ANNs trained on the original data. Analyzing this table carefully, we can see that we get the highest R ($R = 0.8787$ and $R = 0.9292$) for the training set for the network with the highest number of neurons in the hidden layers, i.e., 48-24-16-8-4. However, the RMSE, representing the level of overall agreement between the target and modeled dataset, are the smallest for the ANN 24-16-8-4-2. The CE also supports the preference of this ANN. Moreover, the CE for the ANN 48-24-16-8-4 is negative,



(a) (b)

Fig. 2: The measure $MSSDLE$ (Eq. 2) distribution in the 2D city domain for the ANN with hidden layers $24-16-8-4-2$ trained on (a) noised and (b) original data.

which suggests that this network performs worse than the 'no knowledge' model. It means that the specific task of the ANN model, i.e., correct forecasting of the contaminant concentration's spatial and time gradient, is not achieved. Using the same testing dataset, we verified each ANNs quality by the dynamical measures ρ, $MSSDLE_{x,y}$ and $MSDLE$. The time profile of the ρ and $MSSDLE_x$ for a subset of the analyzed ANNs architectures is presented in Fig. 1. The first look at the figure shows that the networks trained on the noised data (solid lines) perform better than those trained on the original data (dashed lines), regardless of the ANN architecture. Moreover, the ρ value is 2-3 times smaller. The reason is that the ANNs can better learn to forecast small concentrations thanks to introducing the noise. Adding the noise after re-scaling gives a diverse set of small numbers representing the close-to-zero concentrations instead a constant one. The noise contribution to the ANN knowledge is greatly seen in Fig. 2 presenting the 2D distribution of the $MSDLE$ (Eq.2) for ANNs with the same architectures but trained on original and noised data. The agreement of the ANN trained using noised data is almost perfect, as far as its values are close to zero for nearly the whole domain. The result is much worse for the ANN with the same architecture but trained using original data. The above-described results conclude that the ANN 24-16-8-4-2 trained using the noised data seems to be the best-trained network among the considered ones.

6 Summary

We have presented the results of training the FFNN to simulate airborne contaminant transport in highly urbanized terrain. The applied ANN structure allowed training the FFNN to simulate the time-dependent nonlinear function in two

DOI: 10.1007/978-3-031-36021-3_35

spatial dimensions. The training dataset consisted of independent vectors representing the point concentration for the assumed release scenario. On the other side, we require the ANN to simulate the contaminant distribution correctly for spatial and time gradients. It occurred that in such a task, classical measures like R and MSE cannot indicate the best ANN reliably. Therefore, we proposed additional measures to verify the quality of the ANN model. Beneficial are the measures estimating the time dynamic of the ANN model like $\rho(d^t_{ANN}, d^t_{target})$, $MSSDLE_{x,y}$ and $MSDLE$. These measures, as the best one, pointed to the ANN 24-16-8-4-2. This network was not the most extensive and was not characterized by the highest R-value in training. We have presented a significant increase in the ANN quality trained using the noised data. The reason was that the small diversity in the re-scaled target values of concentrations allowed the ANNs to fit the weights in the learning process better.

The presented results lead to the conclusion that the application of ANNs in a new field should be followed by a careful analysis of the verification methods and maybe an adaptation of additional measures as a stopping criterion in the ANN training process.

Acknowledgements We acknowledge Michael Brown and Los Alamos National Laboratory for the possibility to use the Quick Urban & Industrial Complex Dispersion Modeling System. This work was partially supported by Ministry of Education and Science, project number: DNK/SP/549572/2022.

References

1. Hutchinson, M., Oh, H., Chen, W.: A review of source term estimation methods for atmospheric dispersion events using static or mobile sensors. Information Fusion **36**, 130–148 (2017)
2. Kopka, P., Wawrzynczak, A.: Framework for stochastic identification of atmospheric contamination source in an urban area. Atmospheric Environment **195**(1), 63–77 (2018) https://doi.org/10.1016/j.atmosenv.2018.09.035
3. Wawrzynczak, A., Berendt-Marchel, M.: Computation of the Airborne Contaminant Transport in Urban Area by the Artificial Neural Network. In: Krzhizhanovskaya, V.-V., Závodszky, G., (eds) Computational Science – ICCS 2020. ICCS 2020. Lecture Notes in Computer Science, vol 12138, pp. 401–413. Springer, Cham. (2020) https://doi.org//10.1007/978-3-030-50417-5$_3$0
4. Wawrzynczak, A., Berendt-Marchel, M.: Feedforward neural networks in forecasting the spatial distribution of the time-dependent multidimensional functions. 2022 International Joint Conference on Neural Networks, IJCNN, pp. 1-8. Padua, Italy (2022). https://doi.org/10.1109/IJCNN55064.2022.9892001
5. Williams, M.-D., Brown, M.-J., Singh, B., Boswell, D.: QUIC-PLUME theory guide. Los Alamos National Laboratory **43**, (2004)
6. Wawrzynczak, A., Berendt-Marchel, M.: Can the Artificial Neural Network Be Applied to Estimate the Atmospheric Contaminant Transport? In: Dimov, I., Fidanova, S. (eds) Advances in High Performance Computing. HPC 2019. Studies in Computational Intelligence vol 902, pp. 132–142. Springer, Cham. (2021) https://doi.org//10.1007/978-3-030-55347-0$_1$2