IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE)
e-ISSN: 2278-1684,p-ISSN: 2320-334X, Volume 14, Issue 3 Ver. V. (May - June 2017), PP 18-32
www.iosrjournals.org

Performance Analysis Of Different Model Architectures Utilized In An Adaptive Neuro Fuzzy Inference System For Contraction Scour Prediction

Minh Duc Bui[1], Keivan Kaveh[1], Peter Rutschmann[1]
[1](Institute of Hydraulic and Water Resources Engineering, Technical University of Munich, Germany)

Abstract: *The processes involved in the local scour due flow contraction are so complex that it is difficult to establish a general empirical analytical model to provide accurate estimation of scour. In this paper, the capacity of an Adaptive-Network-Based Fuzzy Inference System (ANFIS) for predicting equilibrium contraction scour depth in alluvial channels was investigated. The main subject of this work is to design an appropriate neural network architecture for training the ANFIS from a given set of input and output data. The training algorithms used in this work are (1) basic hybrid method, (2) basic backpropagation with gradient descent method, (3) backpropagation with momentum method, and (4) backpropagation with Levenberg-Marquardt method. Applying a self-developed software, the numerical experiments were carried out by combining these training algorithms with different ANFIS structures. Statistical indices of model performance such as mean average error, root mean squared error, and coefficient of correlation were measured for each combination. The results showed that among all given models the zero order Takagi-Sugeno's model with four bell-shaped membership functions for each input and the Levenberg-Marquardt algorithm for training provided the best performance for estimating of contraction scour depth.*

Keywords: *ANFIS, ANN, Contraction Scour, Soft Computing*

I. Introduction

When the overall width of a river segment is significantly reduced, the water velocity and transport capacity of the flow increase. As soon as the bottom shear stress is higher than the so called critical shear stress, the contraction scour starts to develop into depth by bed material transport. Due to progress of the erosion, the flow cross section becomes larger and the bottom shear stress declines until equilibrium conditions are reached. At this equilibrium scour depth the quantity of bed material being transported into the contracted reach is equivalent to that being transported out of the reach. The scour depth mostly is approximated to be at the centerline of the contracted section and refers to the maximum depth of eroded soil. Generally, two different scour depths can be characterized for the contraction scour profile: the maximum and the mean one. The former can be found directly downstream of the contraction and the latter occurs further downstream. Furthermore, two different conditions for contraction scour can be identified: clear-water and live-bed scour. Clear-water conditions describe the case in which the approaching flow is transporting an insignificant amount of sediment. Live-bed circumstances are complied, when the bed materials upstream of the constriction are in motion and being transported into the contracted section.

Over a period of several decades various investigations were carried out, leading to several, mainly empirical formulae for estimating equilibrium maximal depth of contraction scour. Straub proposed a one-dimensional model based on Du Boys sediment transport formula and the Manning´s equation for estimating the equilibrium scour depth (see [1]). Laursen ([2], [3]) adopted Straub's approach and used his own total sediment transport formula instead. Following the scheme proposed by Laursen, most analytical models essentially adopt the same framework, i.e. the local flow constriction was conceptualized as a cross-sectional variation. Different flume tests were conducted to develop and validate the equations for contraction scours. Komura ([4]) obtained a live-bed and clear-water contraction scour equation by applying dimensional analyses to his experimental results on long contractions. The effect of armoring in the contracted section was integrated by the parameter of the geometric standard deviation. Hence, for its high values (heterogeneous bed materials) the influence on the scour depth was considered. Assuming that the sediment transport rate is proportional to the excess shear stress, Gill ([5]) generalized Straub's live-bed contraction scour formula. Lim and Cheng ([6]) derived a formula for live-bed long contraction scour depths similar to Gill ([5]). They simplified the equation by showing that the only solution of the equation was one where the scour depth was affected by the width ratio of uncontracted to contracted flow alone. The comparison with various sets of laboratory data for long contractions showed that their proposed equation, in spite of its simplicity, was acceptable for practice purposes ([6]) under live-bed as well as under clear-water conditions. Richardson and David ([7]) have been developed the equations, which are used in HEC-18 for the scour calculation with homogenous non-cohesive bed materials. Briaud et al. ([8])

performed clear-water long contraction scour experiments in flumes with fine bed materials. Furthermore, it was reported that the scour depth is not influenced by the contraction length for the case ratios of contraction length to contracted width greater than one quarter. Additionally, an influence of the transition angle between approaching flow and narrowed part could be denied based on their results. Dey and Raikar ([9]) conducted a set of flume experiments in a long contraction under clear-water conditions. Their formula accounts for the maximum scour depth for sand and gravel beds with various geometric standard deviations. A significant effect of sediment gradation for non-uniform bed material on the scour depth due to armoring was observed. All the contraction scour equations were simply based on uniform flow using open channel theories. The increased turbulence within the contracted channel was neglected as one of the assumptions. The critical velocity initiating the scour was proposed by different authors for a certain flume condition, but no critical velocity equation covering the most general cases exits. Further, the difference between them was the use of variant types of empirical sediment transport formulas. Hence, all empirical analytical relations are of limited generality. In many cases, unreasonable contraction scour depths are predicted and the results of the different formulae often vary strongly. The reasons are assumed in the complexity of contraction scour and in limitations of the nonlinear regression applied in these empirical methods, which is reasonable to explore new approaches for scour prediction.

In recent years, the methods of artificial neural networks (ANN) and adaptive neuro-fuzzy inference system (ANFIS) provided good results in the fields of hydraulic engineering. In contrast to most traditional empirical methods, which need prior knowledge about the nature of the relationships among the data, these data driven systems learn from data examples presented to them in order to capture the subtle functional relationships among the data even if the underlying relationships are unknown or the physical meaning is difficult to explain. Furthermore, these methods do not need to be introduced with an explicit form of the regarded task and additionally have proven a high tolerance against data sample errors. These attributes make the utilization of artificial neural networks for contraction scour depth predictions very promising.

In this paper we design an optimal ANFIS, which adequately predicts equilibrium scour depths for long contractions in clear-water conditions including the formation of armor layers. For this purpose we analyze the capability and accuracy of numerous ANFIS models trained with different learning rules for prediction of scour depths. To evaluate the prediction qualities of the designed networks, a comparative study is carried out for these models by evaluating several statistical parameters that describe the errors associated with the model in terms of statistical measures of goodness-of-fit between the observed and estimated scour depths.

II. Adaptive neuro fuzzy inference system

Artificial Neural Networks are considered to be a flexible modeling tool capable of learning the mathematical mapping between input and output variables of the nonlinear systems. Applications of ANN in the hydraulic engineering have been presented by numerous authors e.g. Azmathullah et al. ([10]), Alp and Cigizoglu ([11]), Bateni et al. ([12]), Lee et al. ([13]) and others. Most of the ANN applications in water resources engineering involve the employment of conventional feed forward back propagation method (FFBP), which consists of layers of parallel processing elements called neurons, with each layer being fully connected to the preceding layer by interconnection strengths, or weights. Initial estimated weight values are progressively corrected during a training process that compares predicted outputs with known outputs. Learning of these ANNs is generally accomplished by back-propagation (BP) algorithm. The objective of the BP algorithm is to find the optimal weights, which would generate an output vector, as close as possible to the target values of the output vector, with the selected accuracy. The optimal weights are found by minimizing a predetermined error function. The performance of FFBP was found superior to conventional statistical and stochastic methods. However, the FFBP algorithm is rather sensitive to the selected initial weight values and may provide performances differing from each other significantly. Another problem faced during the application of FFBP is the local minima issue. During the training stage, the networks are sometimes trapped by the local error minima preventing them to reach the global minimum.

The fuzzy system (FS) is suitable for knowledge acquisition and handling such fuzzy information as expert’s experience with respect to the observed input−output data. The fuzzy logic system has been widely applied to modeling, control, identification, prediction, etc., however, the fuzzy model lacks a self-learning and adaptive ability. The neural network has been shown to possess a learning and adaptive ability to input−output data. It is proven to have a good approximate capability for a wide range of nonlinear function and has been modeled for nonlinear dynamic systems. However, in system modeling, network training results in a black-box representation. The model developed is difficult to interpret through human language.

The adaptive neuro fuzzy inference system is basically an integration of the techniques of fuzzy systems and artificial neural networks. The ANN provides connectionist structures and learning abilities to the fuzzy systems whereas the fuzzy systems offer ANN a structured framework with high level IF−THEN rule thinking and reasoning. Depending on the types of inference operations upon fuzzy rules, most fuzzy inference systems can be classified into two types: Mamdani's type ([14]) and Takagi-Sugeno's type (TS-type) ([15]). Mamdani's fuzzy inference method is the most commonly seen fuzzy methodology and was among the first control systems built using fuzzy set theory. Mamdani's approach provides the outcome of the fuzzy rule as a fuzzy set for the output variable and hence the step of defuzzification is essential to get a crisp value of the output variable, whereas the TS approach does not require a classical defuzzification procedure and the outcome of the fuzzy rule is a scalar rather than a fuzzy set for the output variable. Hence TS system is more compact and computationally efficient. It is by far the most popular candidate for sample-data based fuzzy modeling and lends itself to the use of adaptive techniques ([15]). The main problem associated with the TS fuzzy logic modeling is related to the selection of the parameters. Therefore, an effective method is required to tune the membership functions in order to minimize the error measures.

Model structure

The ANFIS model proposed by Jang ([16]) maps inputs through input membership functions (MFs) and associated parameters, and then through output MFs to outputs. The membership functions and rules for the fuzzy inference system can be designed by employing the neural training process, which approximate a desired data set. Five layers are used to create this inference system. Each layer involves several nodes described by a node function. Each node is characterized by a node function with fixed or adjustable parameters. The output signals from nodes in the previous layers will be accepted as the input signals in the present layer. After manipulation by the node function, the output signals in the present layer will serve as input signals for the next layer. An exemplary ANFIS network structure with the TS fuzzy logic modeling for one output and two-inputs with four membership functions and four rules is illustrated in Fig.1. Here, square nodes, named adaptive nodes, represent that the parameter sets in these nodes are adjustable. Whereas, circle nodes, named fixed nodes, represent that the parameter sets are fixed in the system.

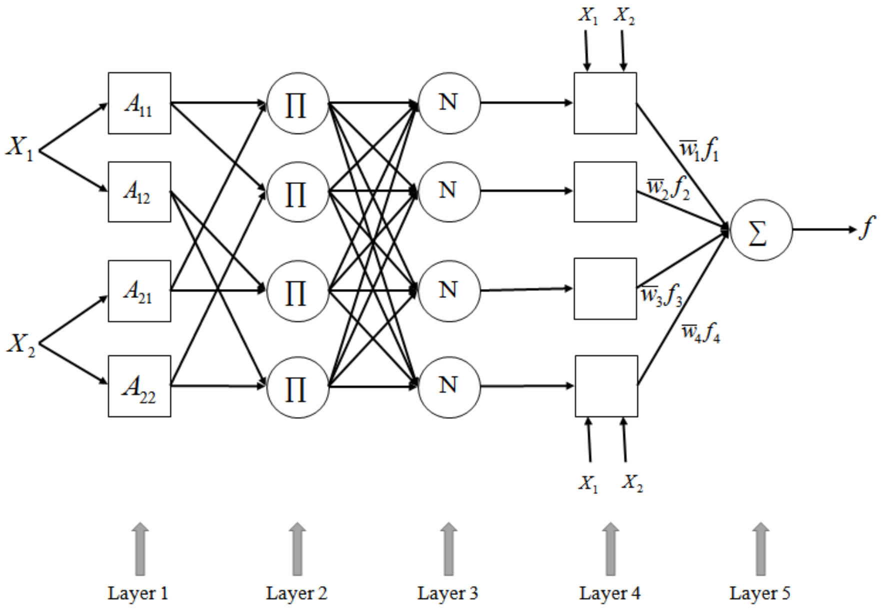

Figure 1: ANFIS architecture with Takagi-Sugeno's model for one output and two-inputs with four membership functions and four rules

In first-order TS's system for two inputs, a typical rule set with four fuzzy rules and four membership functions can be expressed as follows ([1]):

- Rule 1: If x_1 is A_{11} and x_2 is A_{21}, then $f_1 = p_1x_1 + q_1x_2 + r_1$
- Rule 2: If x_1 is A_{11} and x_2 is A_{22}, then $f_2 = p_2x_1 + q_2x_2 + r_2$
- Rule 3: If x_1 is A_{12} and x_2 is A_{21}, then $f_3 = p_3x_1 + q_3x_2 + r_3$
- Rule 4: If x_1 is A_{12} and x_2 is A_{22}, then $f_4 = p_4x_1 + q_4x_2 + r_4$

where x_1 and x_2 are inputs; f_j ($j=1\div 4$) are output levels. For a zero-order TS's model, the output level f_j is a constant ($p_j=q_j=0; j=1\div 4$). ANFIS contain five layers with the following processes:

Layer 1: Each node in this layer produces membership grades of an input variable. The output of the *j*-th node in layer l is denoted as O_j^1. Assuming a generalized bell shaped function as the membership function, the output O_j^1 can be computed as:

$$O_j^1 = \begin{cases} \mu_j(x_1) = \dfrac{1}{1+\left(\left(x_1 - c_j\right)/a_j\right)^{2N_j}}; & j = 1,2 \\ \mu_j(x_2) = \dfrac{1}{1+\left(\left(x_2 - c_j\right)/a_j\right)^{2N_j}}; & j = 3,4 \end{cases} \tag{1}$$

Where $\{a_j, c_j, N_j\}$ are adaptable variables known as premise parameters. The outputs of this layer are the membership values of the premise part.

Layer 2: Every node in this layer multiplies the incoming signals as (**Π**):

$$\begin{aligned} O_1^2 &= w_1 = \mu_1(x_1)\times\mu_3(x_2), \quad O_2^2 = w_2 = \mu_1(x_1)\times\mu_4(x_2) \\ O_3^2 &= w_3 = \mu_2(x_1)\times\mu_3(x_2), \quad O_4^2 = w_4 = \mu_2(x_1)\times\mu_4(x_2) \end{aligned} \tag{2}$$

Layer 3: The *j*-th node of this layer calculates the normalized firing strengths as (**N**):

$$O_j^3 = \overline{w}_j = \frac{w_j}{\sum_{i=1}^{4} w_i}, \quad j = 1 \div 4 \tag{3}$$

Layer 4: Node *j* in this layer calculate the contribution of the *j*-th rule towards the model output, with the following node function:

$$O_j^4 = \overline{w}_j f_j = \overline{w}_j\left(p_j x_1 + q_j x_2 + r_j\right), \quad j = 1 \div 4 \tag{4}$$

Where $\overline{w}_j$ is the output of layer 3, and $\{p_j, q_j, r_j\}$ are parameters set. Parameters of this layer are referred to as consequent.

Layer 5: The single node in this layer calculates the overall output of the ANFIS as (**Σ**):

$$O_1^5 = \sum_{i=1}^{4} \overline{w}_i f_i = f \tag{5}$$

More details about the ANFIS approach can be found in [17].

Learning algorithms

Learning or training phase of a neural network is a process to determine parameter values to sufficiently fit the training data. Once a network structure has been designed for a particular application, the learning process or training can be implemented. This essentially consists of three main parts: (1) compute the outputs; (2) compare the outputs with the desired target values; and (3) adjust the weights and repeat the process. Learning algorithms are used to modify these weights in an orderly fashion to achieve a desired objective of accuracy. Currently there are more than a hundred learning algorithms available for different network configurations and applications. One of the most widely used algorithms in the field of water resources and environmental engineering is the basic Backpropagation algorithm, which seeks to minimize the measure of error, usually defined as the sum of squared differences between the network's outputs and desired ones. For the case with only one output, we have:

$$E = \frac{1}{2}\sum_{p=1}^{P}\left(e^{(p)}\right)^2 \quad ; \quad e^{(p)} = f^{(p)} - t^{(p)} \tag{6}$$

where $f^{(p)}$ is the calculated output for training pattern *p*; $t^{(p)}$ is the desired target output for training pattern *p*; $e^{(p)}$ is the error for training pattern *p*; and *P* is the number of training patterns. In every training algorithm, an attempt is made to reduce the global error ***E*** by adjusting the weights. The basic Backpropagation method is essentially a gradient descent process wherein the weights are adjusted in a particular way. Some modifications of this method have been proposed to speed up the convergence process.

For the ANFIS architecture, the task of the learning algorithm is to adjust all the adaptable parameters, namely, the premise parameters $\{a_i, b_i, c_i\}$ and the consequent parameters $\{p_j, q_j, r_j\}$, to construct the ANFIS output match the training data. In all ANFIS learning algorithms, each epoch comprises of a forward pass and a backward pass. The learning algorithms are distinguished from each other by varying the parameters of the activation functions. In the following parts, four different learning methods used in this work for contraction scour prediction are described briefly.

Basic Backpropagation Method

We split the set of total parameters $\boldsymbol{S}$ into two subsets: $\boldsymbol{S_p}$ is the set of premise parameters and $\boldsymbol{S_c}$ the set of consequent parameters. Using the basic Backpropagation learning algorithm, in the forward pass, for a given input pattern and initial values of all parameter set $\boldsymbol{S}$, the actual output of the model is computed directly from Eq.5. In the backward pass, the error signal resulting from the difference between the actual output and the desired output of the model is propagated backward and these parameters are adjusted using the error correction rule. Training of network i.e. error correction is stopped when the value of the error function E has become sufficiently small below a certain threshold. One of the most important problems in learning is the prevention of over fitting. This issue can be addressed by observing the error index of testing data during the learning iterations. The learning algorithm will be terminated when the error index of the testing data starts to increase on average. Prevention of over fitting is the most common way to provide high generalization.

The error function describes the error when approximating or classifying the training data as a function of the parameters of the activation functions. To find the minimum of this function, we are using the Gradient descent method, where the parameters are updated on pattern by pattern basis, until the complete set of training data was utilized for the training of the network. The parameter update is equal to the slope of the error function with an opposite sign to the sign of the gradient of error. The update rate is further scaled by a learning rate η, which controls the speed at which we do the error correction or decides for the rate at which the network learns. The parameter θk of the set of total parameters S is updated using the following equation:

$$\theta_k^{(n+1)} = \theta_k^{(n)} + \eta\left(\Delta\theta_k\right)^{(n)} \quad ; \quad \left(\Delta\theta_k\right)^{(n)} = -\left(\frac{\partial E}{\partial \theta_k}\right)^{(n)} \tag{7}$$

where n is the presentation number i.e. the number of times the system has been presented a pattern or the iteration number. As the basic Backpropagation algorithm depends on the gradient of the error function, the adjustments for flat spots in the error surface may be very small. Therefore, a slow and time consuming progress in error minimization is the result. The speed of convergence can be improved by increasing the learning rate η. However, with a large value of η, the learning algorithm can overshoot the error minimum resulting in oscillations of the weight values between relatively poor solutions. Another way of improving the speed of convergence is to apply the so-called Hybrid method or some modifications of the basic Backpropagation method (e.g. Backpropagation with momentum and Levenberg-Marquardt approaches).

Basic Hybrid Method

The most well-known learning algorithm for ANFIS is hybrid algorithm, where:

- In the forward pass, node outputs values go forward until layer 4 and the consequent parameters belonging to the subset $\boldsymbol{S_c}$ are identified by the least squares method.
- In the backward pass, the output errors are propagated backward and the premise parameters are updated by gradient descent method mentioned above, with θ_k being the parameter of the set of premise parameters $\boldsymbol{S_p}$.

Using this method, the consequent parameters identified are optimal under the condition that the premise parameters are fixed. Generally, the hybrid approach converges faster than the basic Backpropagation approach, since it reduces the dimension of the search space of the original back-propagation method. A detailed description of this algorithm can be found in [17] and [18].

Backpropagation with Momentum Method

Another way of improving the rate of convergence is the inclusion of a momentum to the gradient expression. Therefore, a part of the previous parameter change is added to the current parameter change. Due to the momentum, the Backpropagation can respond to recent trends in the error surface. This helps to smoothen the gradient descent path by ignoring local anomalies in the error surface. Hence, prevention against extreme changes in the gradient is added to the algorithm. The parameter θ_k of the set of total parameters $\boldsymbol{S}$ is now updated using the following equation:

$$\theta_k^{(n+1)} = \theta_k^{(n)} + \eta\left(\Delta\theta_k\right)^{(n)} + \mu\left(\Delta\theta_k\right)^{(n-1)} \tag{8}$$

where μ is the momentum coefficient. If the direction of the gradient remains constant, the algorithm will take increasingly large steps. This improves the convergence rate. Also in a flat plateau, the momentum can decelerate the decrease (due to the flat gradient) in weight adjustment. Another aspect is that the momentum can help to escape from local minima of the error function. This is useful when the learning rate η alone is not able to generate large enough steps to overcome the ridge of the minima "valley". Of course, if the learning starts in deep local minima (due to the random weight initialization) the momentum does not help to find the way out. However, in general the momentum increases the chances of the back-propagation in finding the global minima of the error surface.

Backpropagation with Levenberg-Marquardt Method

According to Jang and Mizutani ([19]) ANFIS is a network architecture that allows systematic calculations of gradient vectors (derivatives of output error with respect to modifiable parameters), so we are not limited to the back-propagation or hybrid learning method only. In fact, we can apply any gradient-based techniques in nonlinear regression and optimization for network learning. The update rule for Levenberg-Marquardt can be presented as:

$$\begin{aligned} &\theta^{(n+1)} = \theta^{(n)} + (\Delta\theta)^{(n)} \quad ; \quad \Delta\theta = \left(J^T J + \lambda I\right)^{-1} J^T e \\ &\theta = (\theta_k ; k \in S) \quad ; \quad \Delta\theta = (\Delta\theta_k ; k \in S) \quad ; \quad e = \left(e^{(p)} \; ; \; p = 1,\ldots,P\right) \end{aligned} \tag{9}$$

where ***I*** is the identity matrix; θ is the parameter vector to be updated; ***J*** is the Jacobian matrix containing the first derivatives of the global error function E in respect to the parameters; ***e*** is the error row-vector; and λ is a positive constant. Depending on the magnitude of λ, the method transits smoothly between two extremes: the Gauss-Newton method (when $\lambda \rightarrow 0$) and gradient descent method (when $\lambda \rightarrow \infty$). Usually, the Gauss-Newton method is more efficient but less stable and the gradient descent method more stable but less efficient. By properly setting the value of λ, the Levenberg-Marquardt method can be efficient as well as stable. Normally, the damping factor λ is adjusted at each iteration. If reduction of the error function ***E*** is rapid, a smaller value can be used, bringing the algorithm closer to the Gauss–Newton algorithm, whereas if an iteration gives insufficient reduction in the residual, λ can be increased, giving a step closer to the gradient descent direction. A detailed description of this algorithm can be found in [19].

III. Model setup

An ANFIS largely removes the requirement for manual optimization of the fuzzy system parameters. A neural network with a suitable learning method is used to automatically tune the parameters of the membership functions, leading to significantly improved performance of the neuro-fuzzy system. In a neural network, the training essentially builds the system. However when using an ANFIS model, the system is built by fuzzy logic definitions and eventually refined using neural network training algorithms. The three steps involved in the ANFIS model setup are (1) definition of the model input and output; (2) generation of the data set; and (3) design of the network system.

Model input and output

We idealize a rectangular long contraction channel at an equilibrium scour condition shown in Fig.2. where d_s is the equilibrium scour depth (m); L is the length of contraction (m); h_1 is the approaching flow depth; h_2 is the flow depth in contracted part (m); b_1 is the approaching channel width (m); and b_2 is the contracted channel width (m). The channel is considered as a long contraction one, when the ratio $L/b_1 > 1$ (see [4]).

In the context of sediment transport, armoring describes a process by which an erosion resistant surface layer is formed. This occurs in mixed-sediment bed channels, when the forces of the flowing water removes finer particles and leave relatively large particles behind. The coarser bed material builds up a natural riprap-like armor layer over a heterogeneous mixture of sediment with a wide range of diameter. Usually, this sub-layer has a thickness of one or two particle diameters. This armoring limits the further scour development for a particular flow rate. In some cases, the formed armor layer can be destroyed again by higher water discharges, leading to a further development of the scour until a new armor layer is formed or the maximum scour is reached. Applying of formulae which have been developed for uniform bed materials can lead to an overestimation of the equilibrium scour depth of mixed-sediment bed. To find out whether an armor layer is going to form or not, the values of the geometric standard deviation (σ_g) can be examined. According to Gessler ([20]) armoring will arise for a geometric standard deviation greater than two. Later Raudkivi and Ettema ([21]) reported that the bed material will not armor for values of σ_g smaller than one and a half.

Dimensional analysis is a mathematical technique that determines the relationship between several variables. That way, physical phenomena can be formulated as relation between a set of dimensionless groups of variables. The number of these groups is less than the number of dimensional variables. Hence, for the evaluation of contraction scour formulas the dimensional analysis provides a useful tool. The physical parameters influencing the scour depth in a long contraction are given in a functional form in following equation:

$$d_s = F_1\left(u_1, h_1, \rho, \rho_s, g, v, d_m, b_1, b_2, \sigma_g\right) \tag{10}$$

where u_1 is the approaching flow velocity (m/s), ρ is the density of water (kg/m^3), ρ_s is the density of sediment (kg/m^3), g is the acceleration of gravity (m/s^2), ν is the kinematic viscosity of water (m^2/s), and d_m is the median sediment particle size (m).

According to Dey and Debnath ([9]), the independent variables g, ρ, and ρ_s can be represented by one combined parameter Δg as follows:

$$\Delta g = \left(\frac{\rho_s}{\rho} - 1\right) g \tag{11}$$

According to Yallin ([22]) the kinematic viscosity of fluid is considered negligible for a fully turbulent flow over a rough bed, as is assumed in the following. By using the Buckingham theorem for the dimensional analysis with the repeating variables approaching flow velocity u_1 and width b_1, one obtains the functional form of the non-dimensional contraction scour depth $\overline{d_s}$ represented in the following equation:

$$\overline{d_s} = F_2\left(\overline{d}, \overline{Fr}, \overline{h}, \overline{b}, \sigma_g\right) \quad ; \quad \overline{Fr} = \frac{u_1}{\sqrt{\Delta g d_m}} \tag{12}$$

where $\overline{b} = b_2/b_1$ is the channel opening ratio that shows the influence of geometric contraction on the degree of contraction scour; $\overline{d} = d_m/b_1$ represents the impact of sediment size on scour depth; $\overline{h} = h_1/b_1$ refers to the importance of approaching flow depth on scour depth; σ_g indicates the effect of sediment gradation accounting also for armoring of mixed sediment bed; and $\overline{Fr}$ is the densimetric Froude number.

Based on Eq.12, the inputs used for ANFIS contraction scour modeling are five non-dimensional variables of the function $\boldsymbol{F_2}$ and the only output is the non-dimensional maximum equilibrium contraction scour depth $\overline{d_s}$.

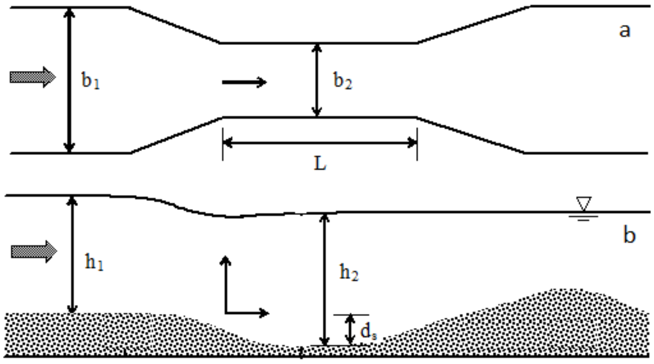

Figure 2: Schematic of a long rectangular channel contraction at equilibrium scour conditions; (a) -top view; (b) - side view, ([23])

Data collection

In the present paper, the experimental data conducted by Dey and Raikar ([23]) and the data of other investigators ([4]; [5]; [24]; and [25]) for long contraction scours in clear-water conditions are used. The whole data set consists of 182 samples with the relevant parameters for contraction scour. In Table 1 the ranges of the different parameters are listed. They also represent the applicable domain of the developed model. Further, since the geometric standard deviation has a range between 1.0 and 3.60, in some cases an armor layer could be formed during the scouring time. The data set has been divided randomly into three subsets, whereby the biggest amount of data (70%) is added randomly to the training subset. The remaining data set samples are used for testing and validating the networks.

To achieve a better performance and faster learning for the network, all data are normalized. The normalized range after the data preprocessing is chosen to be between minus one and one (i.e. ra_{min}= -1 and ra_{max}= 1). To ensure this, formula (13) is utilized. For a simpler and more understandable comparison of the computed outputs with the targets, a post-processing is included as well. Here the outputs and targets are denormalized again, thus set back to the ranges before preprocessing. For this purpose, formula (14) is applied.

$$x_{pr} = \frac{(ra_{\max} - ra_{\min})(x - x_{\min})}{(x_{\max} - x_{\min})} + ra_{\min} \tag{13}$$

$$x_{po} = \frac{(x_{pr} - ra_{\min})(x_{\max} - x_{\min})}{(ra_{\max} - ra_{\min})} + x_{\min} \tag{14}$$

where x_{pr} is the preprocessed variable, x_{po} is the postprocessed variable, x is the original variable, and ra is the normalized range.

Table 1: Range of data parameters

Limit	d_m(mm)	u_1(m/s)	h_1(m)	b_1(m)	b_2(m)	σ_g	$\overline{Fr}$	d_s(m)
Lower	0.350	0.1932	0.0240	0.400	0.100	1.00	1.1434	0.0100
Upper	14.25	0.9290	0.1366	1.586	0.524	3.60	3.2882	0.1626

Network design

The ANFIS architecture used in the work contains a five-layer feedforward neural network and implements TS fuzzy inference system for a systematic approach to generating fuzzy rules from a given input output dataset. The combination of fuzzy inference to represent knowledge in an interpretable manner and the learning ability of neural network that can adjust the membership functions parameters and linguistic rules directly from data enhances the system performance.

Altogether 96 different ANFIS models were configured with three different types of membership functions: Gaussian, Bell-shaped, triangular and trapezoidal functions for inputs as well as zero-order and first-order Sugeno's functions for output. The number of MFs for inputs ranged from two to four. Four different learning approaches presented in the part 2.2 were applied for numerical experiments. The maximal number of training process was 300 epochs.

In order to start the training process, initial values for all model parameters had to be set. These parameters have been defined empirically by trial and error and using the performance measure. The type of all membership functions has been then chosen at the end of the training process.

The creation of fuzzy rules is usually completed by an expert based on his experience. The partitioning of inputs into a degree of membership functions and valid combination of inputs to give typical values for the outputs usually takes a lot of time to design a collection of fuzzy rules. There are various techniques which can automate this process and significantly reduce the development time while improving performance and interpretability of the fuzzy model. The simple way to partition the inputs to form the antecedents of fuzzy rules is by implementing equalized universe method ([16]). Using this method, the input spaces are equally partitioned with the chosen number of clusters and the membership functions are set equal spaces along the range of each data dimension. After the cluster numbers have been specified and the equalized universe method performed to the dataset, the membership functions for each input can be obtained. To generate a set of rules, the grid partition method has to be implemented. The grid partition method works by enumerating all possible combinations of membership functions of all inputs. For example, a fuzzy model with two inputs and three membership functions assigned to each input would result in 3^2=9 fuzzy rules. Fig.3 shows the association of each input membership functions that generate nine rules. In the present study, we used also this method to create fuzzy rules.

The performance of the proposed models can be evaluated using several statistical parameters that describe the errors associated with the model in terms of statistical measures of goodness of fit. In order to provide an indication of goodness-of-fit between the observed and estimated values, the root mean square error (*RMSE*), mean absolute error (*MAE*), and correlation coefficient (*R*) were calculated and evaluated. The *RMSE* evaluates the variance of errors independently from the sample size, which is given by the following equation:

$$RMSE = \sqrt{\frac{1}{P}\sum_{p=1}^{P}\left(f^{(p)} - t^{(p)}\right)^2} \tag{15}$$

The smaller the *RMSE*, the better is the performance of the model. The mean absolute error represents the mean of all the individual errors and indicates whether the model overestimates or underestimates the dependent variable. This statistic is calculated by the following equation:

$$MAE = \frac{1}{P}\sum_{p=1}^{P}\left|f^{(p)} - t^{(p)}\right| \tag{16}$$

The correlation coefficient a concept from statistics is a measure of how well trends in the predicted values follow trends in past actual values. It is a measure of how well the predicted values from a forecast model fit with the observed data and obtained via the following equation:

$$R = \frac{\sum_{p=1}^{P}\left(f^{(p)} - \overline{f}\right)\left(t^{(p)} - \bar{t}\right)}{\sqrt{\sum_{p=1}^{P}\left(f^{(p)} - \overline{f}\right)^{2}\sum_{p=1}^{P}\left(t^{(p)} - \bar{t}\right)^{2}}} \tag{17}$$

$$\overline{f} = \frac{1}{P}\sum_{p=1}^{P} f^{(p)} \quad ; \quad \bar{t} = \frac{1}{P}\sum_{p=1}^{P} t^{(p)}$$

Essentially, the higher the *R* value (with 1 being the maximum value), the better the performance of the model. Briefly, the models predictions are optimal if *R* is found to be close to itself and *RMSE* as well as *MAE* close to zero.

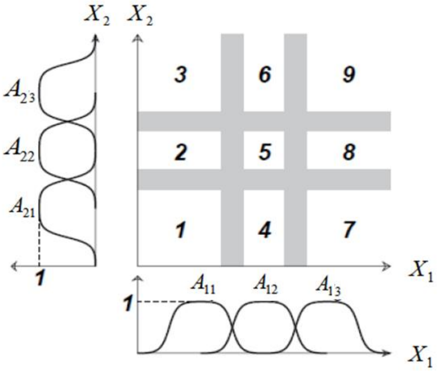

Figure 3: Schematic equalized universe method into nine fuzzy regions

Using MATLAB, the method of either back propagation or combination of least squares estimation and Backpropagation can be applied to estimate and modify membership function parameters. Since other training methods are not included in the current version of the MATLAB, a program packet written in FORTRAN has been developed. When possible, more efficient learning algorithms were incorporated into the current software.

IV. Result and discussion

The parameters to be optimized in ANFIS are the premise parameters which describe the shape of the membership functions, and the consequent parameters which describe the overall output of the system. The optimum parameters obtained are then used in testing session to predict the scour depth. A number of 136 datasets were utilized during training session and 46 datasets were used during testing and validating sessions.

The basic model then varied in four different types of MFs for inputs, two different types of MFs for output and four different training methods. Depending upon the model and its combination, the ANFIS system produced different results. The performance of various models against the measured data was assessed quantitatively in terms of performance indices. The Table 2 shows the tabulation of the statistical performance of the different models applied for the testing data set. The performance indicators in throughout the study are Correlation Coefficient (*R*), Root Mean Square Error (*RMSE*) and Mean Absolute Error (*MAE*). As it can be seen from this table, the ANFIS system is sensitive to number of membership function. Adding further membership functions to the system did not always improve the result. For those models, increasing the number of membership functions from two to three improved the results. However in most cases, increasing the number of membership functions from three to four would lead to the performances decreasing. In Table 2, the best statistical values for each learning method are bolded. It can be concluded that the best combination of the type and number of MFs depends on the choice of the learning algorithm. For the Basic Hybrid learning algorithm, the ANFIS model that utilized two generalized bell MFs with constant Sugeno's function provides the best results. For ANFIS models trained with the Basic Backpropagation technique, the model with three triangular MFs and linear Sugeno's function provides the best performance. Applying Backpropagation with Momentum method, the ANFIS model with two triangular membership functions and linear Sugeno's function provides the best performance. The network trained by Lenenberg-Marquardt's approach performs best with four generalized bell MFs and constant Sugeno’s function. Scatter plots are displayed in Fig.4 for these best performance models applied on the whole data set. The figure shows the observed scour depth on the *x*-axis against the simulated scour depth on the *y*-axis. In each of the scatter plots, a perfect estimation was placed on the 1:1 line. The ANFIS estimated scour depths were distributed evenly on both sides of the line of perfect agreement and fell relatively close to this line. It is evident that all these ANFIS models simulated the contraction scour depth with a good accuracy. Fig.5 shows exemplarily the convergence curves of network error for the best performance models mentioned above in the training period. As it is obvious from this figure, the Levenberg-Marquardt’s method provided the best speed up for the training process. Using the Basic Backpropagation method, a large number of epochs may be needed to adapt the parameters of the MFs.

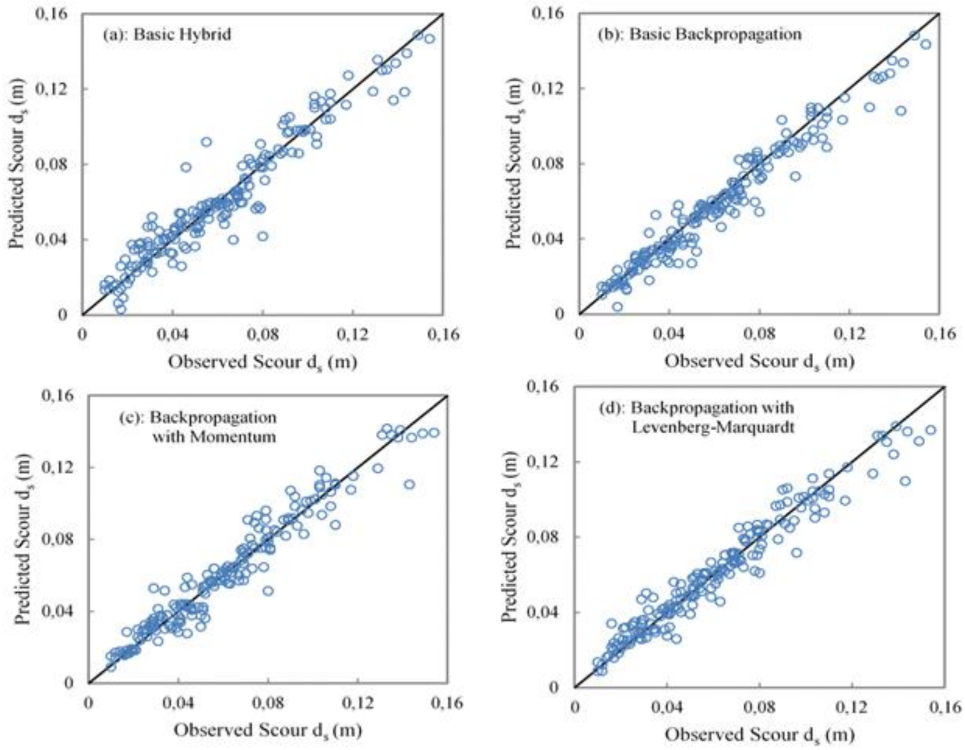

Figure 4: Comparison between the experimental data and the equilibrium scour depths computed with different learning methods in ANFIS model

The mean values of performance indices of the models, processed with different type of MFs and for the testing data set, are listed in Table 3 and Table 4 for comparison purposes. It was found that the contraction scour depths predicted vary to relatively great extent between the various MFs. Using the Gaussian MF or the bell shaped MF for the inputs and zero-order Sugeno’s function for the output, the best predictions were obtained. The remaining MFs still performed worse than the above-mentioned MFs. The worst MF is the trapezoidal MF.

Table 5 shows the averages of statistical indices of the models using different learning algorithms and for the testing data set. The results confirm again that the capability of the ANFIS model in predicting the contraction scour depth using the Levenberg-Marquardt’s algorithm is considerably better than those using other learning methods.

To evaluate the performance of the ANFIS network in predicting contraction scour depth, a comparative study for various models for long contraction scour under clear water condition has also been made where the best ANFIS network trained by Lenenberg-Marquardt’s approach with four generalized bell MFs and constant Sugeno’s function was found. A comparison between this ANFIS model to four of the existing formulae, which have been proposed by Komura ([4]), Lim and Cheng ([6]), Richardson and Davis ([7]), Dey and Raikar ([23]), as well as the ANN model developed by Bui et al. ([1]), was performed using the same 182 observed data set. Fig.6 and Table 6 illustrate the correlation for these prediction methods with the statistical performance indices between predicted and observed data. This table and figure indicate that the performance of the designed ANFIS model is the best among the other prediction models under considerations.

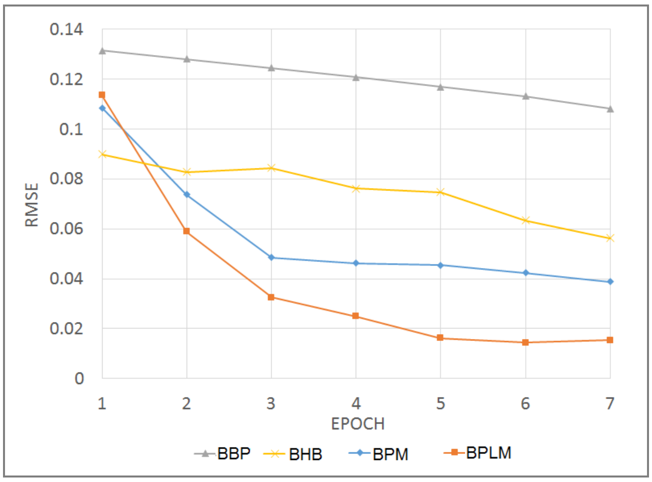

Figure 5: Network error convergence using different learning methods (BBP – Basic Backpropagation; BHB – Basic Hybrid; BPM – Backpropagation with Momentum; BPLM – Backpropagation with Levenberg-Marquardt)

Table 2: Performances of the different ANFIS models for the testing data set

	MF	SF	MFNr	R	RMSE	MAE
Basic Hybrid	gauss	con	2	0.9702	0.0067	0.0048
			3	0.9128	0.0128	0.0091
			4	0.5541	0.0290	0.0181
		lin	2	0.9446	0.0106	0.0087
			3	0.6788	0.0250	0.0162
			4	0.4233	0.0488	0.0381
	gbell	**con**	**2**	**0.9729**	**0.0061**	**0.0043**
			3	0.9586	0.0063	0.0053
			4	0.7227	0.0113	0.0111
		lin	2	0.9439	0.0108	0.0081
			3	0.7372	0.0288	0.0130
			4	0.5144	0.0368	0.0238
	tri	con	2	0.9612	0.0100	0.0068
			3	0.9170	0.0139	0.0105
			4	0.5477	0.0339	0.0221
		lin	2	0.8644	0.0186	0.0144
			3	0.8845	0.0152	0.0115
			4	0.4882	0.0473	0.0307
	trap	con	2	0.9633	0.0080	0.0060
			3	0.2962	0.1048	0.0323
			4	0.3702	0.1269	0.0461
		lin	2	0.8335	0.0176	0.0130
			3	0.7653	0.0196	0.0117
			4	0.4129	0.0569	0.0298

Note: MF – Type of membership function (gauss – Gaussian, gbell – bell shaped, tri – triangular, trap – trapezoidal); SF – Sugeno's function (con – constant or zero-order TS, lin – linear or first order TS); MFNr – number of membership function; R – correlation coefficient; RMSE – root mean square error; MAE – mean absolute error.

Table 2: Performances of the different ANFIS models for the testing data set (cont.)

	MF	SF	MFNr	R	RMSE	MAE
Basic Backpropagation	gauss	con	2	0.9798	0.0080	0.0054
			3	0.9755	0.0085	0.0066
			4	0.9677	0.0086	0.0067
		lin	2	0.9802	0.0066	0.0059
			3	0.9769	0.0072	0.0053
			4	0.9697	0.0081	0.0062
	gbell	con	2	0.9761	0.0075	0.0060
			3	0.9726	0.0096	0.0071
			4	0.9736	0.0090	0.0059
		lin	2	0.9758	0.0076	0.0049
			3	0.9707	0.0081	0.0064
			4	0.9684	0.0093	0.0074
	tri	con	2	0.9780	0.0102	0.0056
			3	0.9681	0.0105	0.0064
			4	0.9382	0.0107	0.0086
		lin	2	0.9806	0.0084	0.0061
			3	**0.9811**	**0.0069**	**0.0050**
			4	0.9344	0.0107	0.0085
	trap	con	2	0.9719	0.0073	0.0060
			3	0.9327	0.0112	0.0072
			4	0.6460	0.0370	0.0300
		lin	2	0.9305	0.0126	0.0107
			3	0.9165	0.0119	0.0079
			4	0.8368	0.0198	0.0144

Note: MF – Type of membership function (gauss – Gaussian, gbell – bell shaped, tri – triangular, trap – trapezoidal); SF – Sugeno's function (con – constant or zero-order TS, lin – linear or first order TS); MFNr – number of membership function; R – correlation coefficient; RMSE – root mean square error; MAE – mean absolute error.

Table 2: Performances of the different ANFIS models for the testing data set (cont.)

	MF	SF	MFNr	R	RMSE	MAE
Backpropagation with Momentum	gauss	con	2	0.9811	0.0072	0.0048
			3	0.9804	0.0081	0.0060
			4	0.9725	0.0079	0.0062
		lin	2	0.9806	0.0076	0.0055
			3	0.9809	0.0068	0.0057
			4	0.9756	0.0074	0.0060
	gbell	con	2	0.9636	0.0083	0.0067
			3	0.9403	0.0114	0.0113
			4	0.9197	0.0132	0.0100
		lin	2	0.9712	0.0116	0.0094
			3	0.9634	0.0093	0.0067
			4	0.9561	0.0105	0.0076
	tri	con	2	0.9812	0.0069	0.0053
			3	0.9703	0.0087	0.0067
			4	0.9592	0.0085	0.0072
		lin	**2**	**0.9821**	**0.0061**	**0.0057**
			3	0.9815	0.0062	0.0058
			4	0.9560	0.0101	0.0066
	trap	con	2	0.9680	0.0087	0.0064
			3	0.9476	0.0096	0.0067
			4	0.9344	0.0115	0.0092
		lin	2	0.9656	0.0095	0.0078
			3	0.9664	0.0078	0.0063
			4	0.9192	0.0125	0.0088

Note: MF – Type of membership function (gauss – Gaussian, gbell – bell shaped, tri – triangular, trap – trapezoidal); SF – Sugeno's function (con – constant or zero-order TS, lin – linear or first order TS); MFNr – number of membership function; R – correlation coefficient; RMSE – root mean square error; MAE – mean absolute error.

Table 2: Performances of the different ANFIS models for the testing data set (cont.)

Backpropagation with Levenberg-Marquardt	MF	SF	MFNr	R	RMSE	MAE
	gauss	con	2	0.9818	0.0068	0.0051
			3	0.9799	0.0071	0.0056
			4	0.9772	0.0073	0.0056
		lin	2	0.9820	0.0070	0.0052
			3	0.9819	0.0088	0.0066
			4	0.9752	0.0071	0.0057
	gbell	**con**	2	0.9818	0.0059	0.0045
			3	0.9567	0.0093	0.0069
			4	**0.9824**	**0.0056**	**0.0037**
		lin	2	0.9699	0.0077	0.0060
			3	0.9771	0.0091	0.0065
			4	0.9341	0.0116	0.0087
	tri	con	2	0.9800	0.0096	0.0065
			3	0.9717	0.0117	0.0076
			4	0.9708	0.0073	0.0060
		lin	2	0.9815	0.0068	0.0054
			3	0.9723	0.0084	0.0062
			4	0.8386	0.0183	0.0134
	trap	con	2	0.9670	0.0075	0.0061
			3	0.9746	0.0071	0.0058
			4	0.9595	0.0081	0.0066
		lin	2	0.9500	0.0107	0.0094
			3	0.9627	0.0102	0.0076
			4	0.9354	0.0116	0.0093

Note: MF – Type of membership function (gauss – Gaussian, gbell – bell shaped, tri – triangular, trap – trapezoidal); SF – Sugeno's function (con – constant or zero-order TS, lin – linear or first order TS); MFNr – number of membership function; R – correlation coefficient; RMSE – root mean square error; MAE – mean absolute error.

Table 3: Averages of performance indices for different types of membership function for the testing data set

Type	R	RMSE	MAE
Gaussian	0.9201	0.0112	0.0083
Bell shaped	0.9251	0.0110	0.0080
Triangular	0.9162	0.0127	0.0091
Trapezoidal	0.8469	0.0229	0.0127

Table 4: Averages of performance indices for two types of Sugeno's function for the testing data set

Type	R	RMSE	MAE
Constant	0.9182	0.0120	0.0087
Linear	0.8860	0.0169	0.0103

Table 5: Averages of performance indices for different learning methods for the testing data set

Method	R	RMSE	MAE
Backpropagation with Levenberg-Marquardt	0.9643	0.0088	0.0067
Backpropagation with Momentum	0.9632	0.0090	0.0070
Basic Backpropagation	0.9459	0.0106	0.0079
Basic Hybrid	0.7349	0.0294	0.0165

Table 6: Performance indices of various approaches for the whole data set

	ANFIS	ANN	Richardson and Davis	Dey and Raikar	Komura	Lim and Cheng
R	0.976	0.965	0.853	0.836	0.790	0.748
RMSE	0.013	0.015	0.054	0.035	0.170	0.048
MAE	0.005	0.006	0.028	0.013	0.084	0.016

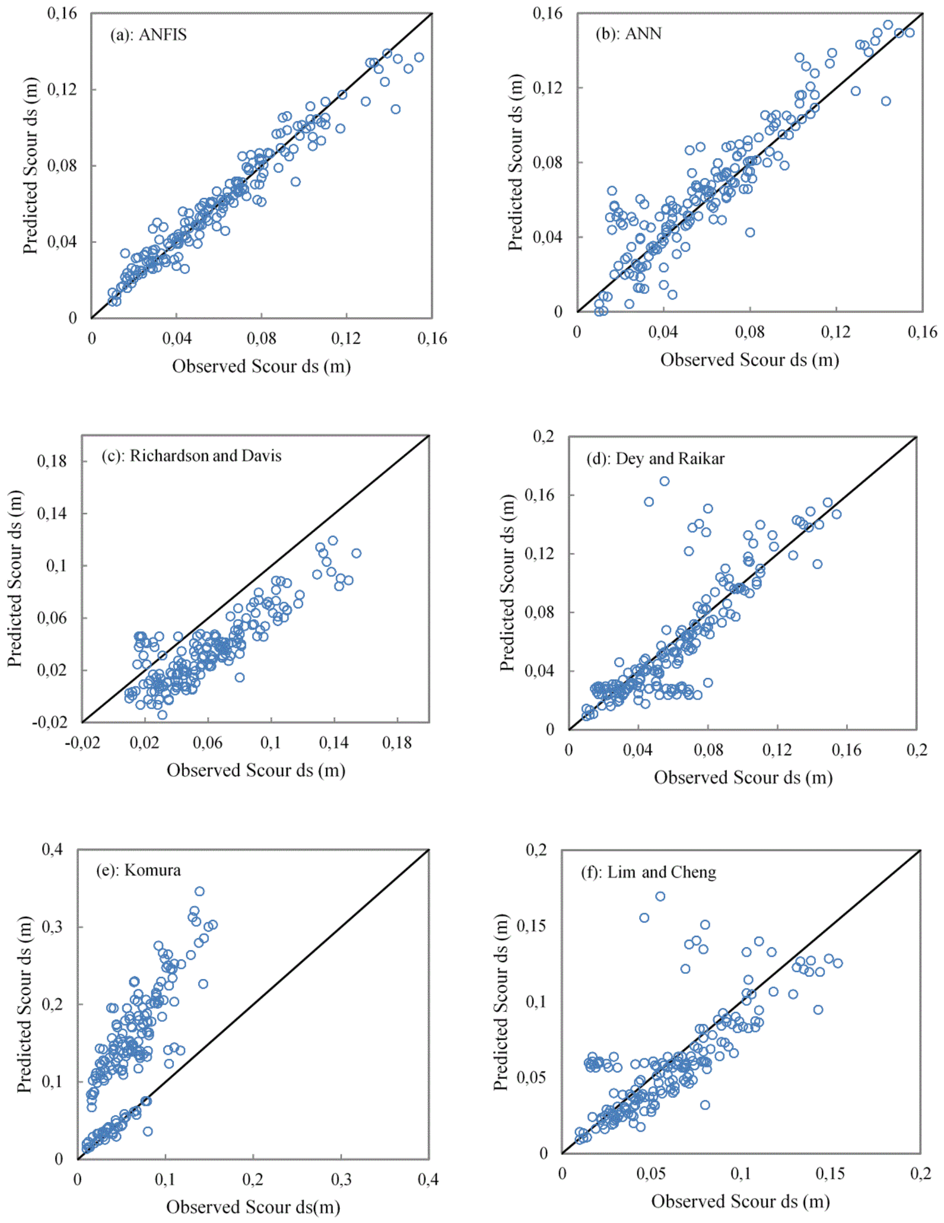

Figure 6: Comparison between the experimental data and the equilibrium scour depths predicted with different methods

V. Conclusion

The main subject of this work is to analysis the capability of different learning algorithms with different ANFIS architectures for contraction scour prediction. A FORTRAN based program system using different learning methods for Takagi-Sugeno's model has been developed. The data sets of laboratory measurements were compiled from published literature and used to train the ANFIS networks. The trained networks were validated by using the observations that were not involved in training. Based on statistical indices of model performance, we conclude that the most relevant parameters for ANFIS in scour depth prediction problem are the input membership function number and membership function shape. Furthermore, comparison between four algorithms - namely basic hybrid method, basic back propagation, back propagation with momentum method, and back propagation with Levenberg-Marquardt method – was made to update the membership functions. Based on the performance indices, the four best model structures for each training methods were selected, which predicted the contraction scour depths considerably more precisely than those predicted by ANN or other conventional methods. Among them, the ANFIS network trained by Lenenberg-Marquardt's approach performs best with four generalized bell MFs and constant Sugeno's function.

The numerical experiments point out ANFIS has great ability in predicting complex and non-linear phenomena, especially in situations where the relationships among physical processes were not fully understood. In this study, after an extensive trial and error search the optimal ANFIS network for maximum equilibrium contraction scour modeling under clear water conditions was found. However, as stated also by some of the authors, the relative performance of the networks is greatly influenced by the particular problems.

References

[1] M.D. Bui, K. Keivan, P. Penz, and P. Rutschmann, Contraction scour estimation using data driven methods, *J. of Applied Water Engineering and Research. DOI: 10.1080/23249676.2015.1051141*, 2015.

[2] E.M. Laursen, Scour at bridge crossings, *J. of Hydraulic Engineering, ASCE, 86(2)*, 1960, 1-54.

[3] E.M. Laursen, An analysis of relief bridge scour, *J. of Hydraulic Division, ASCE, 89(3)*, 1963, 93-118.

[4] S. Komura, Equilibrium depth of scour in long constrictions, *J. of Hydraulic Division, ASCE, 92(5)*, 1966, 17-38.

[5] M.A. Gill, Bed erosion in rectangular long contraction, *J. of Hydraulic Division, ASCE, 107(3)*, 1981, 273-284.

[6] S.Y. Lim, N.S. Cheng, Scouring in long contractions, *J. of Irrigation and Drainage Engineering, ASCE, 124(5)*, 1998, 258-261.

[7] E.V. Richardson, S.R. Davis, *Evaluating scour at bridges* (Washington, D.C.: Hydraulic Engineering Circular No. 18, Federal Highway Administration, 2001).

[8] J.L. Briaud, H.C. Chen, Y. Li, P. Nurtjahyo, and J. Wang, SRICOS-EFA Method for Contraction Scour in Fine-Grained Soils, *J. of Geotechnical and Geoenvironmental Engineering, ASCE*, 2005, 1283-1294.

[9] S. Dey, K. Debnath, Sediment pickup on streamwise sloping beds, *J. of Irrigation and Drainage Engineering, 127(1), ASCE*, 2001, 39-43.

[10] H.M. Azmathullah, C.K. Chang, A.A. Ghani, J. Ariffin, N.A. Zakaria, and Z.A. Hasan, An ANFIS-based approach for predicting the bed load for moderately sized rivers, *J. Hydro-envir. Research, 3*, 2009, 35-44.

[11] M. Alp, H.K. Cigizoglu, Suspended sediment load simulation by two artificial neural network methods using hydrometeorological data, *Envir. Model and Software, 22*, 2007, 2-13.

[12] S.M. Bateni, S.M. Borghei, and D.S. Jeng, Neural network and neuro-fuzzy assessment for scour depth around bridge piers, *Eng. Appl. Artif. Intell. 20*, 2007, 401-414.

[13] T.L. Lee, D.S. Jeng, G.H. Zhang, and J.H. Hong, Neural network modeling for estimation of scour depth around bridge piers, *J. Hydrodynamics. Ser. B, 19(3)*, 2007, 378-386.

[14] E.H. Mamdani, Application of fuzzy algorithms for simple dynamic plant, *Proc. IEEE 121*, 1974, 1585–1588.

[15] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, *IEEE Transactions on Systems, Man and Cybernetics, 15*, 1985, 116–132.

[16] J.S.R. Jang, ANFIS: adaptive-network-based fuzzy inference system, *IEEE Transactions on Systems, Man and Cybernetics, 23(3)*, 1993, 665–685.

[17] J.S.R. Jang, C.T. Sun, Neuro-fuzzy modelling and control, *Proc. IEEE*, 1995, 378–406.

[18] J.S.R. Jang, C.T. Sun, and E. Mizutani, *Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence* (Upper Saddle River, New Jersey, Prentice Hall, 1997).

[19] J.S.R. Jang, E. Mizutani, Levenberg-Marquardt method for ANFIS learning, in: Fuzzy Information Processing Society, NAFIPS, Biennial Conference of the North American, *Proc. IEEE*, 1996, 87-91.

[20] J. Gessler, Critical shear stress for sediment mixtures, *Proc. 14th Cong. of IAHR, Vol. 3*, 1971, C1_1-C1_8.

[21] A. Raudkivi, R. Ettema, Stability of armor layers in rivers, *J. of Hydraulic Division, ASCE, 108(9)*, 1982, 1047-1057.

[22] M.S. Yallin, *Mechanics of sediment transport* (New York: Pergamon, 1997).

[23] S. Dey, R.V. Raikar, Scour in Long Contractions, *J. of Hydraulic Engineering, ASCE, 131(12)*, 2005, 1036-1049.

[24] M.G. Webby, General scour at contraction, *RRU Bulletin 73, National Roads Board, Bridge Design and Research Seminar, New Zealand*, 1984, 109-118.

[25] S.Y. Lim, S. Lim, Clear water scour in long contractions, *Proceedings of the ICE - Water Maritime and Energy, 101(2)*, 1993, 93-98.