Ocean Engineering 38 (2011) 995–1000

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

Linear genetic programming to scour below submerged pipeline

H.Md. Azamathulla [a,*], Aytac Guven [b], Yusuf Kagan Demir [b]

[a] River Engineering and Urban Drainage Research Centre (REDAC), Universiti Sains Malaysia, Engineering Campus, Seri Ampangan, 14300 Nibong Tebal, Pulau Pinang, Malaysia
[b] Department of Civil Engineering, University of Gaziantep, 27310 Gaziantep, Turkey

ARTICLE INFO

Article history:
Received 8 September 2010
Accepted 19 March 2011
Available online 13 April 2011
Editor-in-Chief: A.I. Incecik

Keywords:
Local scour
Linear genetic programming
Neuro-fuzzy
Pipelines

ABSTRACT

Genetic programming (GP) has nowadays attracted the attention of researchers in the prediction of hydraulic data. This study presents Linear Genetic Programming (LGP), which is an extension to GP, as an alternative tool in the prediction of scour depth below a pipeline. The data sets of laboratory measurements were collected from published literature and were used to develop LGP models. The proposed LGP models were compared with adaptive neuro-fuzzy inference system (ANFIS) model results. The predictions of LGP were observed to be in good agreement with measured data, and quite better than ANFIS and regression-based equation of scour depth at submerged pipeline.

1. Introduction

Scour due to current and wave action is a major cause for the failure of underwater pipelines. Interactions between the pipeline and its erodible bed under strong current and/or wave conditions may cause scour around the pipelines, leading to partial or even complete burial of pipeline. This process involves the complexities of both the three-dimensional flow pattern and sediment movement. Scour underneath the pipeline may expose a section of the pipe, causing it to become unsupported. If the free span of the pipe is long enough, the pipe may experience resonant flow-induced oscillations, leading to settlement and potentially structural failure. Accurate estimate of the scour depth is important in the design of submarine pipelines (Chiew, 1991). The estimation of the scour characteristics of underwater pipelines continues to be a concern for hydraulic engineers.

A number of empirical formulas have been developed in the past to estimate equilibrium scour depth below pipelines, including Chao and Hennessy (1972), Kjeldsen et al., (1973), Ibrahim and Nalluri (1986), Bijker and Leeuwestein (1984), Moncada-M and Aguirre-Pe (1999), and Chiew (1991). However, the main deficiency of these formulas is that the empirical equations do not model actual scour process. A summary of these traditional equations is given in following section.

Predictive approaches such as artificial neural networks (ANN) (Kisi et al., 2009) and adaptive neuro-fuzzy inference systems (ANFIS; Azamathulla et al., 2008) have been recently shown to yield effective estimates of scour around hydraulic structures. ANNs have been reported to provide reasonably good solutions for hydraulic-engineering problems, particularly for cases of highly nonlinear and complex relationship among the input–output pairs in corresponding data (Azmathullah et al., 2006; Kisi et al., 2009; Bilhan et al., 2010).

During last two decades, researchers have noticed that the use of soft computing techniques as alternative to conventional statistical methods based on controlled laboratory or field data, yields significantly better results. ANN and GP are the most widely used branches of soft computing in hydraulic engineering. Within the larger field of hydraulics, several researchers have dealt with the scour around and downstream of hydraulic structures using ANN (Azmathullah et al., 2005, 2006, 2010; Guven and Gunal, 2008a) and GP (Guven and Gunal, 2008b; Guven et al., 2009; Azamathulla and Ab. Ghani, 2010). Linear genetic programming (LGP), which is an extension to GP, recently has attracted the attention of researchers in prediction of hydraulic characteristics (Guven, 2009; Guven and Aytek, 2009).

This study presents LGP and ANFIS as alternative tools in the prediction of scour below pipeline. The objective of this study is to develop a predictive model for scour depth using LGP. The performance of the proposed LGP model is compared with an ANFIS and conventional regression-based equations. The explicit formulation of the LGP model is also presented.

2. Analysis of local scour below underwater pipelines

The variables influencing the equilibrium scour depth (d_s) below a pipeline in a steady flow over a bed of uniform, spherical, and cohesionless sediment as shown in Fig.1 are flow condition,

* Corresponding author. Tel.: +60 45995867; fax: +60 45941036.
E-mail addresses: redacazamath@eng.usm.my, mdazmath@gmail.com (H.Md. Azamathulla), aguven@gantep.edu.tr (A. Guven), ykdemir@gantep.edu.tr (Y.K. Demir).

0029-8018/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.oceaneng.2011.03.005

Notation

ρ	fluid density,
ρ_s'	buoyant sediment density,
v	fluid kinematic viscosity,
Q	discharge,
Y	flow depth,
g	gravitational acceleration,
d_{50}	particle mean diameter,
S_f	slope of the energy line,
D	the diameter of the pipe,
d_s	equilibrium scour depth,
F_r	Froude number,
R_e	Reynolds number,
V	average flow velocity,
α	spread,
τ_*	dimensionless Shields parameter,
R^2	coefficient of determination,
RMSE	root mean squared error,
MAE	mean average error,
δ	average absolute deviation.

sediment characteristics, and pipe geometry. The scour depth can be represented by the following general functional relationship (Moncada-M and Aguirre-Pe, 1999):

$$d_s = f(\rho, \rho_s', v, Q, Y, g, d_{50}, S_f, D) \quad (1)$$

where ρ=fluid density; ρ_s'=buoyant sediment density; v=fluid kinematic viscosity; Q=discharge; Y=flow depth; g=gravitational acceleration; d_{50}=particle mean diameter; S_f=slope of the energy line; D=the diameter of the pipe; and d_s=equilibrium scour depth.

The nine independent variables in Eq. (1) can be reduced to a set of six non-dimensional parameters. The Buckingham pi (or π) theorem applied to Eq. (1), choosing ρ, Q, and D as basic variables, leads to

$$\frac{d_s}{D} = f\left(\tau_*, \frac{Y}{D}, \frac{D}{d_{50}}, F_R, R_e\right) \quad (2)$$

where τ_*=dimensionless shields parameter related to sediment transport; D/d_{50}=dimensionless soil characteristics; $F_R = V/\sqrt{gY}$, Froude number; and $R_e = VD/v$, Reynolds number, where V=average flow velocity. The experimental data were collected from several references such as Moncada-M and Aguirre-Pe (1999) and Dey and Singh (2008). The whole data set consists of 215 data sets. Table 1 shows the range of variation of collected data for this study and its parameters. A summary of the traditional equations is given herein

$$d_s = 0.9722\left(\frac{U_0^2}{2g}\right)^{0.2} D^{0.8} \quad \text{(Kjeldsen et al., 1973)} \quad (3)$$

$$\frac{d_s}{D} = 4.706\left(\frac{U_0}{U_c}\right)^{0.89}\left(\frac{U_o}{g_y}\right)^{1.48} + 0.06 \text{ clearwater}$$

$$\frac{d_s}{D} = 0.084\left(\frac{U_0}{U_c}\right)^{-0.8}\left(\frac{U_o}{\sqrt{g_y}}\right)^{-0.16} + 1.33 \text{ livebed} \quad \text{(Ibrahim and Nalluri, 1986)} \quad (4)$$

$$d_s = 0.929\left(\frac{U_o}{2g}\right)^{0.26} D^{0.79} d_{50}^{-0.04} \quad \text{(Bijker and Leeuwestein, 1999)} \quad (5)$$

$$\frac{d_s}{D} = 0.9\tanh(1 + 1.4F) + 0.55,$$

$$\frac{d_s}{D} = 2F\sec\left(1.7\frac{e}{D}\right) \quad \text{(Moncade-M and Aguirre-Pe, 1999)} \quad (6)$$

where D is the pipe diameter (m) and e is the initial gap between pipe and undisturbed erodible bed (m).

2.1. Linear genetic programming (LGP)

LGP, which is an extension to conventional tree-based GP, evolves developing sequences of instructions from an imperative programming language (C or C++) or from a machine language clarify. The

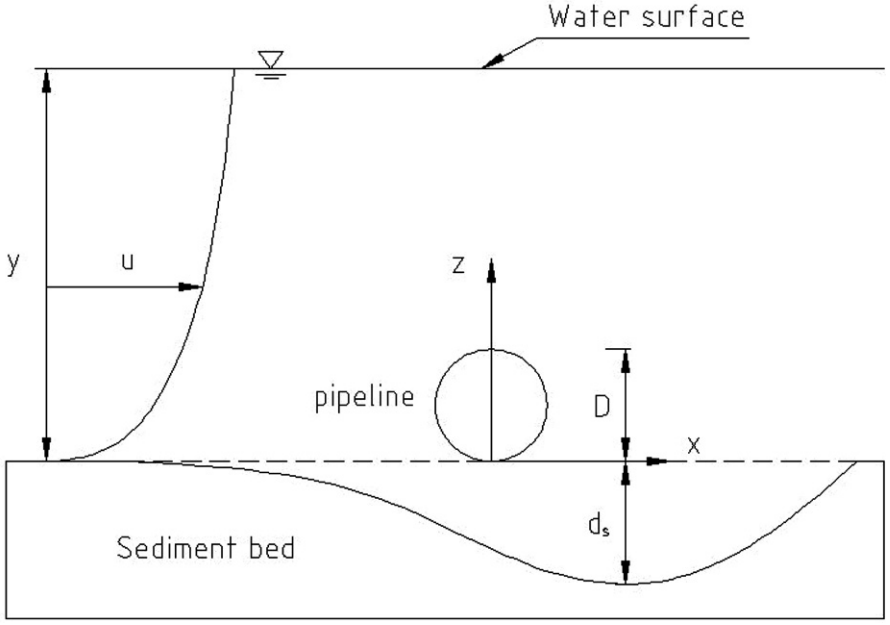

Fig. 1. Local scour below pipeline (Dey and Singh, 2008).

Table 1
Data variation.

Parameters	Unit	Data range	Mean	Std dev
(a)Range of different input–output parameters used for the estimation of scour depth				
Flow discharge (Q)	**cm³/s**	7–94.42	35.11	21.74
Flow depth (Y)	**cm**	3.8–28	13.43	6.21
Particle mean diameter (d_{50})	**cm**	0.234–0.7	0.437	0.144
Diameter of the pipe (D)	**cm**	0.48–7.6	1.92	1.61
Equilibrium scour depth (d_s)	**cm**	0.02–11.3	4.75	2.39
(b)Range of different non-dimensional input–output parameters used for the estimation of scour depth				
Dimensionless shields parameter (τ_*)		0.038–0.70	0.23	0.17
Normalized flow depth (Y/D)		1.06–7	3.14	1.2
Pipeline diameter cross section of sediment size (D/d_{50})		3.28–145.8	38.17	31.41
Froude number (F_r)		0.2–0.83	0.46	0.15
Reynolds number R_e is normally used		700–9450	3250	2174
Non-dimensional equilibrium scour depth ($-$)		0.008–1.66	1.04	0.32

name "linear" refers to the structure of the (imperative) program representation, and does not reflect functional genetic programs that are restricted to a linear list of nodes only. On the contrary, genetic programs normally represent highly nonlinear solutions (Brameier, 2004). The main differences to conventional tree-based GP are the graph-based data flow that results from a multiple usage of indexed variable (register) contents and the existence of structurally ineffective code (introns; Brameier, 2004; Brameier and Banzhaf, 2001). This concept was expanded to the Automatic Induction of Machine code by Genetic Programming Technique (AIMGP), in which the solutions are directly computed, as binary machine codes and executed without using an interpreter; thus in this way the computer program can evolve very quickly (Bhattacharya et al., 2001; Brameier and Banzhaf, 2001; Foster, 2001).

Each individual program in LGP is represented by a variable-length sequence of simple C language instructions. These instructions operate on one or more registers ($r[i]$) or constants (c) from predefined sets (Oltean and Groşan, 2003; Brameier, 2004). An example of LGP program is

```
Void LGP
  double v[3];
  {
  r[0]+=v[1];
  r[1]-=r[0];
  r[0]/=v[1];
  r[2]+=v[3];
  r[0]*=-0.992;
  r[0]/=r[2];
  }
```

where $v[i]$ represents the input and output variables used in LGP modeling.

The *function set* of the system can be composed of arithmetic operations ($+,-$, /, *), conditional branches (if $v[i]<=v[k]$), and function calls $f \in \{e^x, x, \sin, \cos, \tan, \log, sqrt, \ln, power\}$. Each function implicitly includes an assignment to a variable $v[i]$, which facilitates the use of multiple program outputs in LGP, whereas in tree-based GP those side effects need to be incorporated explicitly (Brameier and Banzhaf, 2001). The functional set and operational parameters used in LGP modeling during this study are given in Table 2.

LGP utilizes two-point string crossover such as in medical data mining (Brameier and Banzhaf, 2001). A segment of random position and random length is selected in both parents and exchanged between them. If one of the resulting children would exceed the maximum length, crossover is abandoned and restarted by exchanging equalized segments (Brameier and Banzhaf, 2001).

Table 2
Parameters of the optimized GP model.

Parameter	Description of parameter	Setting of parameter
p_1	Function set	$+$, $-$, *, /, $\sqrt{}$, power
p_2	Population size	250
p_3	Mutation frequency (%)	96
p_4	Crossover frequency (%)	50
p_5	Number of replication	10
p_6	Block mutation rate (%)	30
p_7	Instruction mutation rate (%)	30
p_8	Instruction data mutation rate (%)	40
p_9	Homologous crossover (%)	95
p_{10}	Program size	Initial 64, maximum 256

An operand or an operator of an instruction is changed by mutation into another symbol over the same set. LGP also employs a special kind of mutation (called *macro mutation*), which deletes or inserts an entire instruction.

The fitness of a LGP individual may be computed using the equation

$$f = \sum_{j=1}^{N} (|O_j - E_j|) \tag{7}$$

where O_j is the value returned by a chromosome for the fitness case j and E_j is the expected value for the fitness case j.

In LGP, the maximum size of the program is usually restricted to avoid over-growing programs without bound (Brameier and Banzhaf, 2001). In this study, the maximum size of each program has been set to 256, starting with 80 instructions per program. This configuration has been tested for each LGP model and has been experienced to be sufficient to handle the nonlinearity and complexity of processes involved. Finally, the best LGP program was converted into a functional representation by successive replacements of $v[i]$ starting with the last effective instruction (see Eq. (16)).

2.2. ANFIS networks

The Adaptive Neuro-Fuzzy Inference System (ANFIS), first introduced by Jang, 1993, is a universal approximator and, as such, is capable of approximating any real continuous function on a compact set to any degree of accuracy (Jang, 1993). Thus, in parameter estimation, where the given data are such that the system associates measurable system variables with an internal system parameter, a functional mapping may be constructed by ANFIS, which approximates the process of estimation of the internal system parameter.

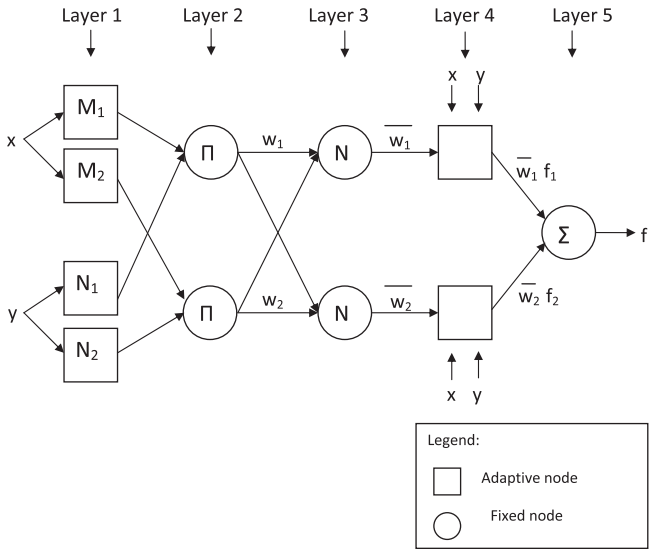

Fig. 2. ANFIS Network Architecture.

Table 3
Sensitivity analysis for independent parameters for the testing set using LGP.

LGP Model-2	RMSE	MAE	R^2
$\frac{d_s}{D}=\Psi\left(\tau_*,\frac{Y}{D},\frac{D}{d_{50}},R_e,F_r\right)$	0.054	0.34	0.88
$\frac{d_s}{D}=\Psi\left(\frac{Y}{D},\frac{D}{d_{50}},R_e,F_r\right)$	0.066	0.46	0.82
$\frac{d_s}{D}=\Psi\left(\tau_*,\frac{D}{d_{50}},R_e,F_r\right)$	0.077	0.56	0.81
$\frac{d_s}{D}=\Psi\left(\tau_*,\frac{Y}{D},R_e,F_r\right)$	0.096	0.75	0.73
$\frac{d_s}{D}=\Psi\left(\tau_*,\frac{Y}{D},\frac{D}{d_{50}},F_r\right)$	0.143	0.85	0.83
$\frac{d_s}{D}=\Psi\left(\tau_*,\frac{Y}{D},\frac{D}{d_{50}},R_e\right)$	0.267	0.95	0.64

The ANFIS is functionally equivalent to fuzzy inference systems (Jang and Sun, 1995). Below, the hybrid learning algorithm (Jang, 1993), which combines gradient descent and the least-squares method, is introduced, and the issue of how the equivalent fuzzy inference system can be rapidly calibrated and adapted with this algorithm is discussed.

Most of the previous works that address ANN applications to water resources have included the feed forward type of the architecture, where there are no backward connections, which are trained using the error back propagation scheme or the FFBP configuration (Azamathulla and Ab. Ghani, 2010). Drawbacks of ANN include that it needs more training time and the difficulties in detecting hidden neurons in hidden layer for better predictions. Therefore, the present study applies a new soft computing technique ANFIS.

The input in ANFIS (Fig. 2) is first converted into fuzzy membership functions, which are combined together. After following an averaging process to obtain the output membership functions, the desired output is finally achieved.

3. Development of LGP and ANFIS models

The following scenarios are considered in building the LGP and ANFIS models with the inputs and output shown in the network. The equilibrium local scour depth (d_s) around a pipeline is influenced by the variables characterizing the flow, bed sediment, and pier geometry, as given in Eq. (1) and relative depth scour in Eq. (2). A sensitivity analysis was done in order to investigate the significance of each in input parameter given right side of Eq. (2) on d_s/D. Initially, a LGP model was developed based on the full-input set given in Eq. (2). Then each parameter was removed from the input set and new LGP models were developed. It was observed that removing any of the input parameters considerably worsened the performance of corresponding LGP model and LGP developed based on the whole input set gave the best performance (see Table 3). This proved the significant influence of each parameter given in right side of Eq. (2) on relative scour depth d_s/D.

The performance of all models was compared using four error measures

$$R^2=1-\frac{\sum_{i=1}^{N}(o_i-t_i)^2}{\sum_{i=1}^{N}(o_i-\bar{o}_i)^2} \tag{8}$$

$$RMSE=\sqrt{\frac{\sum_{i=1}^{N}(o_i-t_i)^2}{N}} \tag{9}$$

$$MAE=\frac{1}{N}\sum_{i=1}^{N}|o_i-t_i| \tag{10}$$

$$\delta=\frac{\sum|(o_i-t_i)|}{\sum o_i}*100 \tag{11}$$

where t_i denotes the target values of relative scour depth (cm/cm), while o_i and $\bar{o}_i$ denote the observed and averaged observed values of relative scour depth (cm/cm), respectively, and N is the number of data points.

3.1. Development of ANFIS model

This network (Fig. 2) works as follows: let x and y be the two typical input values fed at the two input nodes, which will then transform those values to the membership functions (say bell-shaped) and give the output as follows: (Note in general, w is the output from a node, μ is the membership function, and M_i and N_i are fuzzy sets associated with nodes x and y, respectively, in Eq. (12))

$$\mu_{Mi}(x)=\frac{1}{1+|(x-c_1)/a_1|^{2b_1}} \tag{12}$$

where a_1, b_1, and c_1 are changeable premise parameters. Similar computations are carried out for the input of y to obtain $\mu_{Ni}(y)$. The membership functions are then multiplied in the second layer, e.g.

$$w_i=\mu_{Mi}(x).\mu_{Ni}(y)\quad(i=1,2) \tag{13}$$

Such products or firing strengths are then averaged

$$\overline{w}_i=w_i/\sum w_i\quad(i=1,2) \tag{14}$$

Nodes of the fourth layer use the above ratio as a weighting factor. Furthermore, using fuzzy if–then rules produces the following output: (an example of an if–then rule: if x is M_1 and y is N_1, then $f_1=p_1x+q_1y+r_1$)

$$\overline{w}_if_i=\overline{w}_i(p_ix+q_iy+r_i) \tag{15}$$

where p, q, and r are changeable consequent parameters. The final network output f was produced by the node of the fifth layer as a summation of all incoming signals, which is exemplified in the Eq. (15). The parameters like p, q, and r employed in Eq. (15), for each rule of the ANFIS models, and the corresponding rules of the developed ANFIS models (software) could be made available by an E-mail request to the first writer.

A two-step process is used for faster calibrating and to adjust the network parameters to the above network. In the first step, the premise parameters are kept fixed, and the information is propagated forward in the network to layer 4. In layer 4, a least-squares estimator identifies the important parameters. In the

second step, the backward pass, the chosen parameters are held fixed while the error is propagated. The premise parameters are then modified using gradient descent. Apart from the calibration patterns, the only user-specified information required is the number of membership functions for each input. The description of the learning algorithm is given by Jang and Sun (1995).

The following scenario was considered in building the ANFIS model (Fig. 3) with the inputs and output shown in the network. From the collected data sets used in this study, around 75% of these patterns were used for training (chosen randomly until the best calibration performance was obtained), while the remaining patterns (25%) were used for testing, or validating, the ANFIS model. Software was developed to perform the analysis, and can be obtained from the corresponding author.

3.2. Development of LGP model

In this study, four basic arithmetic operators ($+ \ - \ * \ /$) and some basic mathematical functions ($\sqrt{}$, x^2, power) were utilized to get the optimum LGP formulation. The combination of arithmetic operators and functions utilized in this study were observed to give the best results. A large number of generations are tested to find a formula with minimum error. First, the maximum size of each program was assigned as 256, starting with 80 instructions for program. The program is run until there is no longer significant improvement in the testing/validating performance. The simplified analytic form of proposed LGP model is given in Eq. (16), for interested readers who can re-evaluate LGP in their experimental results

$$d_s/D = ((1-T_1)^8-1)^{0.5} - \frac{T_1}{(1-T_1)^4} \tag{16}$$

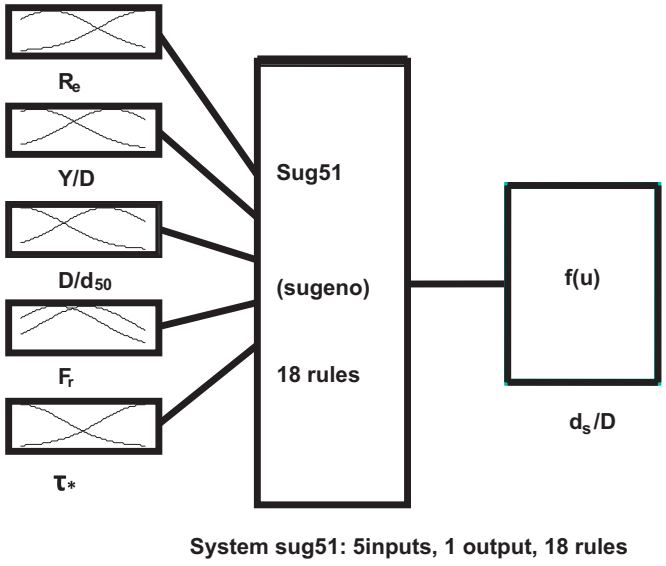

Fig. 3. ANFIS Model-2 for pipeline scour depth.

where

$$T_1 = \left(\frac{(T_2-T_3-T_4-T_5)}{T_6}+T_7\right)\frac{T_9}{(T_9^{0.25}+1.253)^2} \tag{17}$$

$$T_2 = D/d_{50}-4.92\tau_*^{-1}-3.764-\frac{1}{F_R(-1.23\tau_*^{-1}-1)} \tag{18}$$

$$T_3 = \frac{(-4.92\tau_*^{-1}-4-T_1)F_r}{1.048y/DT_2} \tag{19}$$

$$T_4 = 3.012\tau_*^2(T_2-T_3)+1.23\tau_*^{-1} \tag{20}$$

$$T_5 = 0.667(T_4-0.973\tau_*^{-1})-(T_2-T_3-T_4) \tag{21}$$

$$T_6 = \left(\frac{T_5}{y/D}+(T_2-T_3-T_4-T_5)+y/D+D/d_{50}+0.288\right)^2 \tag{22}$$

$$T_7 = \frac{T_6}{D/d_{50}R_e^2}+1.745R_e^{-1} \tag{23}$$

$$T_8 = \frac{0.066(T_7+1.23\tau_*^{-1}-0.176)^2-0.065}{\tau_* y/D((T_2-T_3-T_4-T_5)/T_6+T_7v)^2} \tag{24}$$

$$T_9 = (3.116y/D((T_8+R_e)\tau_*-R_e)^2)^{0.25} \tag{25}$$

T_1– T_9 are the temporary computational programs used in the proposed LGP modeling. The value of the output of these programs is the value remaining in Eq. (16) after the program executes.

4. Results and discussions

The statistical results of model predictions for training and testing sets are given in Table 4. From Table 4, it is clear that LGP model predicted the scour depth for both training and testing set with relatively lower error RMSE (0.039 and 0.045) and higher accuracy (R^2=0.88 and 0.83), respectively. Further, Table 4 also proves the outperforming of the LGP models compared to the ANFIS models and Eq. (2).

For instance, it can be seen from Table 4 that the ANFIS model has smaller RMSE (0.008) and higher R^2 (0.89), compared to the ANFIS (Fig. 4). The underlying fact beneath these results probably is due to a lesser scaling effect. Parallel findings were also addressed by Guven and Gunal (2008a, 2008b) and Azamathulla et al. (2008).

Another interesting observation from Table 4 is that, although the δ results of ANFIS (10.58) and LGP (15.5) for training set are lower than those of LGP model, the MAE values of ANFIS results for testing set are less reliable. Meantime, the testing results of ANFIS models are much worse than those for training and testing (Figs. 4 and 5). This indicates that the ANFIS models has got specialized on the training data and has a poor generalization capacity on the testing data. This issue is known as an over-generalization problem, which is a common issue in neural network techniques (Guven and Gunal, 2008b).

The empirical formula proposed by Kjeldsen et al. (1973) (Eq. (3)) can be said to fail in prediction of both d_s with quite high error (RMSE=3.043, MAE=3.451) and relatively low correlation

Table 4
Comparison of models for non-dimensional set performance of the LGP and ANFIS models.

Models for non-dimensional	R^2		RMSE		MAE		δ	
	Training	Validation	Training	Validation	Training	Validation	Training	Validation
LGP	0.86	0.88	0.039	0.045	0.279	0.480	15.5	14.87
ANFIS	0.89	0.82	0.008	0.035	0.058	0.062	10.58	11.47

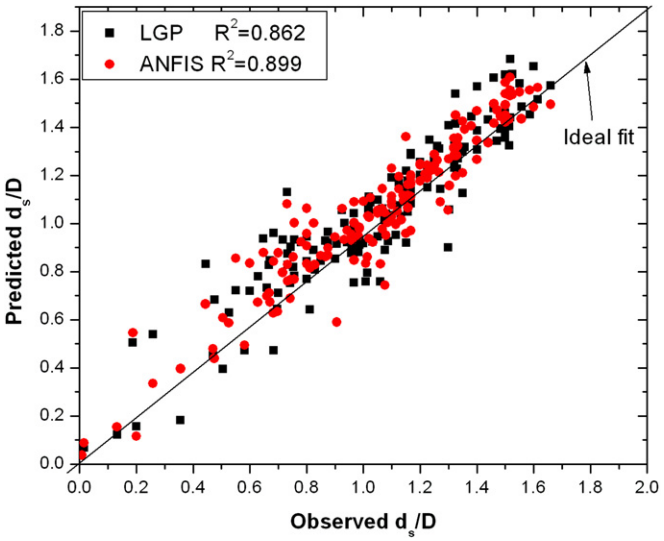

Fig. 4. Scatter plot of observed and predicted relative scour depth using the LGP and ANFIS models for non-dimensional parameter-training.

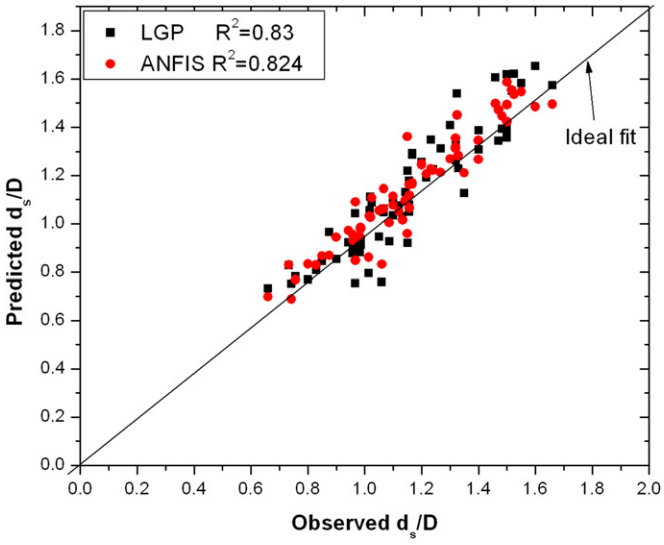

Fig. 5. Scatter plot of observed and predicted relative scour depth using the LGP and ANFIS models for non-dimensional parameter-validation (testing).

($R^2=0.642$), compared to the LGP and ANFIS models. The significant drawback of conventional regression-based equations is the fact that they are restrained by the mathematical shape of the model function, which generally fails to conform to the physical nature of observed data (Guven and Gunal, 2008a, 2008b).

5. Conclusion

The application of the relatively new soft computing approach of genetic programming to predict the local pipeline scour depth was described. The LGP (Eq. 16) and ANFIS models were developed to predict the values of relative scour depth from the laboratory measurements. The equilibrium depth of scour below underwater pipelines was estimated based on optimum data sets including most effective non-dimensional parameters. Application of the LGP in this study is another important contribution to scour-depth estimation methodologies for pipes. The present study indicates that employing the original data set yielded a network that can predict measured depth scour below pipeline more accurately than the standard regression analysis. The overall performance of ANFIS model ($R^2=0.89$) and the LGP model ($R^2=0.86$) were found to be challenging.

Acknowledgments

The first author wishes to express his sincere gratitude to Universiti Sains Malaysia for funding a short term grant to conduct this on-going research (304.PREREDAC.6035262).

References

Azmathullah, H.Md., Deo, M.C., Deolalikar, P.B., 2005. Neural networks for estimation of scour downstream of ski-jump bucket. Journal of Hydraulic Engineering 131 (10), 898–908.
Azmathullah, H.Md., Deo, M.C, Deolalikar, P.B., 2006. Estimation of scour below spillways using neural networks. IAHR, Journal of Hydraulic Research 44 (1), 61–69.
Azamathulla, H.Md., Deo, M.C., Deolalikar, P.B., 2008. Alternative neural networks to estimate the scour below spillways. Advances in Engineering Software 39 (8), 689–698.
Azamathulla, H Md., Ghani, A.A., Zakaria., N.A., Guven, A., 2010. Genetic programming to predict bridge pier scour. ASCE Journal of Hydraulic Engineering 136 (3), 165–169.
Azamathulla, H.Md., Ab. Ghani, A., 2010. Genetic programming to predict river pipeline scour. ASCE, Journal of Pipeline system and Engineering Practice 1 (3), 127–132.
Bhattacharya, M., Abraham, A., Nath, B., 2001. A linear genetic programming approach for modeling electricity demand prediction in Victoria. In: Proceedings of the Hybrid Information Systems, First International Workshop on Hybrid Intelligent Systems, Adelaide, Australia, December 11–12, 379–393.
Bijker, E.W., Leeuwestein, W., 1984. Interaction between pipelines and the seabed under influence of waves and currents. In: Proceedings of the Symposium on International Union of Theoretical Applied Mechanics/International Union of Geology and Geophysics, 235–242.
Bilhan, O., Emiroglu, M.E., Kisi, O., 2010. Application of two different neural network techniques to lateral outflow over rectangular side weirs located on a straight channel. Advances in Engineering Software 41 (6), 831–837.
Brameier, M., 2004 On Linear Genetic Programming. Ph.D. Thesis, University of Dortmund.
Brameier, M., Banzhaf, W., 2001. A comparison of linear genetic programming and neural networks in medical data mining. IEEE Transactions on Evolutionary Computation 5, 17–26.
Chiew, Y.M., 1991. Prediction of maximum scour depth at submarine pipelines. Journal of Hydraulic Engineering 117 (4), 452–466.
Chao, J.L., Hennessy, P.V., 1972. Local scour under ocean outfall pipe-lines. Water Pollution Control Federation 44 (7), 1443–1447.
Dey, S., Singh, N.P., 2008. Clear-water scour below underwater pipelines under steady flow. Journal of Hydraulic Engineering 134 (5), 588–600.
Foster, J.A., 2001. Discipulus: a commercial genetic programming system. Genetic Programming and Evolvable Machines 2, 201–203.
Guven, A., Gunal, M., 2008a. Genetic programming approach for prediction of local scour downstream of hydraulic structures. Journal of Irrigation and Drainage Engineering, ASCE 134 (2), 241–249.
Guven, A., Gunal, M., 2008b. Prediction of scour downstream of grade-control structures using neural networks. Journal of Hydraulic Engineering, ASCE 134 (11), 1656–1660.
Guven, A., Azamathulla, H.Md., Zakaria, N.A., 2009. Linear genetic programming for prediction of circular pile scour. Journal of Ocean Engineering 36 (12–13), 985–991.
Guven, A., Aytek, A., 2009. New approach for stage-discharge relationship: gene-expression programming. Journal of Hydrologic Engineering 14 (8), 812–820.
Guven, A., 2009. Linear genetic programming for time-series modeling of daily flow rate. Earth System Science 118 (2), 137–146.
Ibrahim, A., Nalluri, C., 1986. Scour prediction around marine pipelines. In: Proceedings of the 5th International Symposium on offshore Mechanics and Arctic Engineering, 679–684.
Jang, J.S.R., Sun, C.T., 1995. Neuro-fuzzy modelling and control. Proceedings IEEE 83 (3), 378–406.
Jang, J.S.R., 1993. ANFIS: adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man and Cybernetics 23 (3), 665–685.
Kisi, O., Haktanir, T., Ardiclioglu, M., Ozturk, O., Yalcin, E, Uludag, S., 2009. Adaptive neuro-fuzzy computing technique for suspended sediment estimation. Advances in Engineering Software 40 (6), 438–444.
Kjeldsen, S.P., Gjørsvik, O., Bringaker, K.G., Jacobsen, J., 1973. Local scour near offshore pipelines. In: Proceedings of the 2nd International Conference on Port and Ocean Engineering under Arctic Conditions, Univ. of Iceland, pp. 308–331.
Moncada-M, A.T., Aguirre-Pe, J., 1999. Scour below pipeline in river crossings. Journal of Hydraulic Engineering 125 (9), 953–958.
Oltean, M., Groşan, C., 2003. A comparison of several linear genetic programming techniques. Complex Systems 14 (1), 1–29.