Performance Evaluation of Gene Expression Programming for Hydraulic Data Mining

Khalid Eldrandaly[1] and Abdel-Azim Negm[2]
[1]Information Systems Department, College of Computers, Zagazig University, Egypt
[2]Water Engineering Department, College of Engineering, Zagazig University, Egypt

Abstract: *Predication is one of the fundamental tasks of data mining. In recent years, Artificial Intelligence techniques are widely being used in data mining applications where conventional statistical methods were used such as Regression and classification. The aim of this work is to show the applicability of Gene Expression Programming (GEP), a recently developed AI technique, for hydraulic data prediction and to evaluate its performance by comparing it with Multiple Linear Regression (MLR). Both GEP and MLR were used to model the hydraulic jump over a roughened bed using very large series of experimental data that contain all the important flow and roughness parameters such as the initial Froude number, the height of roughness ratio, the length of roughness ratio, the initial length ratio (from the gate) and the roughness density. The results show that GEP is a promising AI approach for hydraulic data prediction.*

Keywords: *Data mining, multiple linear regression, MLR, gene expression programming, GEP, hydraulic jump.*

Received August 24, 2006; accepted November 23, 2006

1. Introduction

Data mining consists of the extraction of novel, useful and understandable knowledge from observed data [12]. Artificial Intelligence (AI) techniques are being used in a wide variety of data mining applications. They are being used in areas where conventional statistical methods were used such as regression and classification. The problem of regression is usually described as a process of induction of a data model of the system that will be capable of predicting responses of the system that have yet to be observed [11].

Predictions of some hydraulic variables such as the length of the hydraulic jump are highly needed for the design of most hydraulic structures [8]. Hydraulic jump in open channel flow occurs when the fluid passes through a region of critical flow. This region marks the transition of flow from super-critical to sub-critical flow. The hydraulic jump is defined as an abrupt change in the flow depth due to considerable energy losses. Practical applications of hydraulic jump are many; it is used (1) to dissipate energy in water flowing over hydraulic structures and thus prevent scouring downstream from the structures; (2) to recover head or raise the water level on the downstream side of a measuring flume and thus maintain high water level in the channel for irrigation or other water-distribution purposes; (3) to increase weight on an apron and thus reduce uplift pressure under a masonry structure by raising the water depth on the apron; (4) to increase the discharge of a sluice by holding back tailwater; (5) to indicate special flow conditions; (6) to mix chemicals used for water purification; (7) to aerate water for city water supplies; and (8) to remove air pockets from water-supply lines and thus prevent air locking [1]. Detailed description of hydraulic jump is reported elsewhere [1, 5, 10].

The aim of this work is to show the applicability of Gene Expression Programming (GEP), a recently developed AI technique, for hydraulic data prediction and to evaluate its performance by comparing it with Multiple Linear Regression (MLR). Both MLR and GEP, were used to model the hydraulic jump over a roughened bed using very large series of experimental data that contain all the important flow and roughness parameters such as the initial Froude number, the height of roughness ratio, the length of roughness ratio, the initial length ratio (from the gate) and the roughness density.

2. Theoretical Background

Figure 1 shows a definition sketch of a hydraulic jump over roughened bed and the typical arrangement of the roughness on the bed. Practically, it is difficult to derive a theoretical equation for the length of jump [7]. Therefore, the dimensional analysis will be utilized to define the basic factors affecting the length of jump and/or the depth of the hydraulic jump. Equation 1 defines the involved variables in the phenomenon.

$$f(\rho, V_1, y_1, g, \mu, L_j, I, h_b, L_R, x_o) = 0 \qquad (1)$$

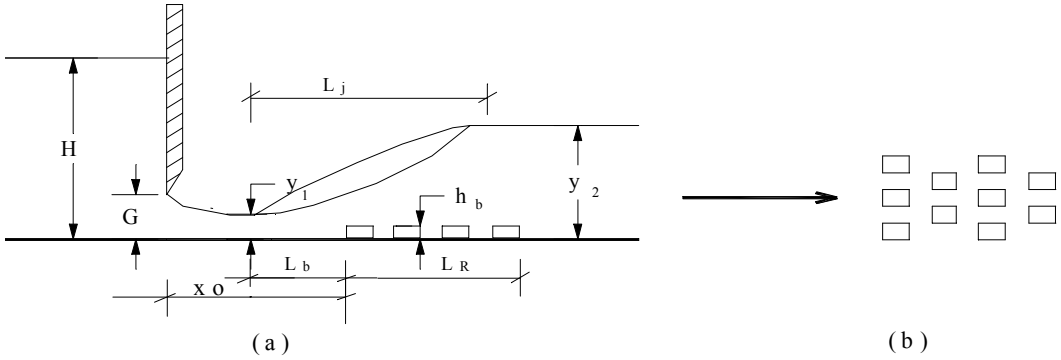

Figure 1. (a) Definition sketch for hydraulic jump over roughened bed (b) Staggered arrangement of roughness patterns.

in which ρ is the density of water, V_1 is the average velocity at the beginning of jump where the depth of flow is y_1, g is the gravitational acceleration, μ is the dynamic viscosity of water, L_j is the length of hydraulic jump over rough bed, L_R, is the length of roughened bed, I is the roughness concentration ($I=100aN/bL_R$, with a being the plan projected area of one roughness element, N is the number of roughness elements and b is the width of the flume), x_o =10+L_b is the distance from the gate to the beginning of the roughness below the jump.

Using the principles of the dimensional analysis, the following relationship is obtained:

$$\frac{L_j}{y_1} = f\left(F_1, R, I, \frac{h_b}{y_1}, \frac{L_R}{y_1}, \frac{x_o}{y_1}\right) \quad (2)$$

In which F_1 is the approaching flow Froude number at the beginning of the jump, R is the Reynolds number based on y_1 which could be neglected as its effect is minor because the viscosity was almost constant. Equation 2 becomes:

$$\frac{L_j}{y_1} = f\left(F_1, I, \frac{h_b}{y_1}, \frac{L_R}{y_1}, \frac{x_o}{y_1}\right) \quad (3)$$

Equation 3 is used to develop the regression model.

3. Experimental Data

The experimental data of the present study were conducted using a horizontal bed flume of rectangular cross section. The flume which is of a recirculation system type has the dimensions of 0.53 m wide, 0.7 m deep and 16.7 overall operating length with 13.92 m working glass walled section. Experimental data were collected from four different experimental setup. Artificial roughness elements were used to roughen the bed in all cases. The cross section of the roughness element is 1.6 cm by 1.6 cm.

In the first experimental setup the roughness height, h_b, and the roughness length, L_R was kept constant to 1.6 cm and 200 cm. Also, L_b is zero and x_o is about 10 cm. Five different roughness intensities were tested in this case, viz I=0, 1.5625, 6.25, 25 and 100%. Figure 2 shows the variation of $L_{j/y1}$ with $F1$ for this data set.

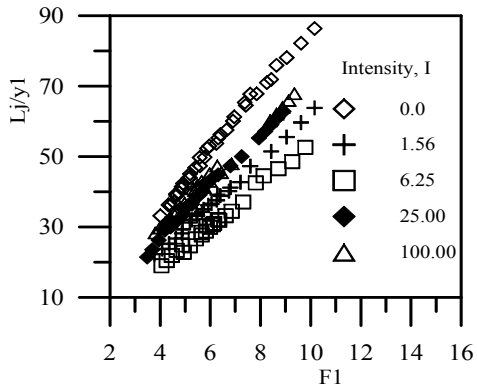

Figure 2. Variation of $L_{j/y1}$ with F1 for first experimental setup (L_R=200cm, L_b=0, x_o=10cm, h_b=1.6 cm).

While in the second experimental setup, the roughness length is kept constant at 66 cm. Also the roughness intensity, I is fixed to 10 cm, L_b is taken zero and x_o is about 10 cm. The roughness height h_b, was varying viz 0.8, 1.2, 1.6 and 2 cm. Figure 3 shows the variation of $L_{j/y1}$ with $F1$ for these data set.

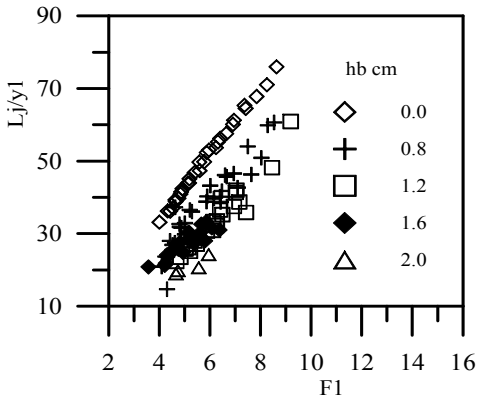

Figure 3. Variation of $L_{j/y1}$ with F1 for second experimental setup (L_R=66cm, I=10% cm, L_b=0, x_o=10cm).

The third experimental setup consisted of varying L_R to 142, 81.1, 49.3, 33.4, 17.5 and 1.6 cm. For each L_R, L_b is varying viz 7, 15, 30, 55, 85 and 140 cm. The roughness height and the roughness intensity were kept constants to 1.6 cm and 10%. Figure 4 shows the

variation of $L_{j/y1}$ with $F1$ for the data set where L_R equals 142 cm for different L_b.

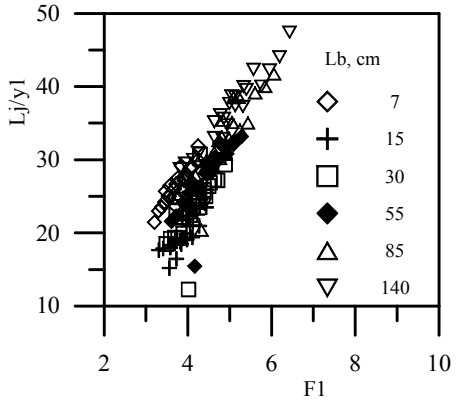

Figure 4. Variation of $L_{j/y1}$ with F1 for one set of the third experimental setup (L_R=142cm, I=10% cm, h_b=1.6cm and x_o=10cm).

The fourth experimental arrangement consisted of testing the effect of L_R and keeping all other independent variables constants. L_R varies from 200.66 to 28.5 cm, I is 10%, h_b is 1.6 cm, L_b is zero and x_o is 10 cm. Figure 5 shows the variation of $L_{j/y1}$ with $F1$ for this data set. The discharge, (Q), was measured using a precalibrated orifice meter to the nearest of ± 0.01 lit/sec. It ranged from about 16 lit/sec to about 60 lit/sec. The water depths of the jump were measured using a point gauge of accuracy of ± 0.01 mm. The length of jump was measured to the nearest of 1 cm. The jump begins at about 10 cm downstream from the sluice gate. The end of the hydraulic jump was taken to be the section at which the sequent depth became equal to the tailwater depth. More details on the experimental investigations of the present study are reported elsewhere in [7].

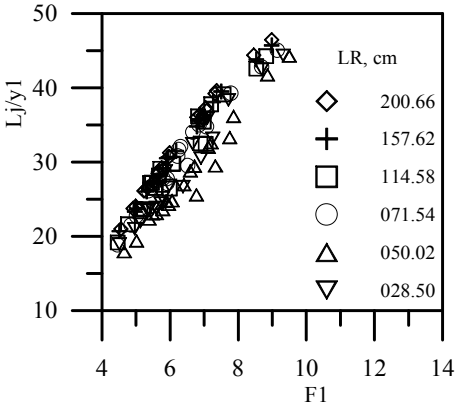

Figure 5. Variation of $L_{j/y1}$ with F1 for one set of the fourth experimental setup (I=10% cm, h_b=1.6 cm, L_b=0, and x_o=10cm).

4. Modeling Hydraulic Jump Using MLR

Many applications of regression analysis involve situations in which there are more than one regressor variable. A regression model that contains more than one regressor variable is called a multiple regression model. MLR makes two critical assumptions. The first is that the outputs and the inputs are linearly related. The second is that there is no interaction between the input variables or attributes and the output. Once this relationship is found, a predicted output for a new set of input variables, not in the original training set, can be computed. Detailed description of MLR is reported elsewhere [6, 9].

In order to develop a general equation for the length of hydraulic jump under various roughness conditions, several trials are attempted employing the multiple linear regression analysis tool of the Neural Connection® 2.1 software package. About 65% of the collected experimental data (about 821 observations) are utilized to build the proposed regression model in the light of equation 3. The rest of the observations (442 observations) are used to validate and test the built model. The following model is obtained.

$$\frac{L_j}{y_1} = +0.111I + 8.191F_1 - 29.387\frac{h_b}{y_1} + 0.050\frac{L_R}{y_1} + +0.202\frac{x_o}{y_1} - 0.4 \quad (4)$$

5. Modelling Hydraulic Jump using GEP

Gene expression programming, an artificial problem solver inspired in natural genotype/ phenotype system, was invented by Ferreria in 1999 [2], and incorporates both the simple, linear chromosomes of fixed length similar to the ones used in genetic algorithms and the ramified structures of different sizes and shapes similar to the parse trees of genetic programming. Thus, the phenotype of GEP consists of the same kind of ramified structure used in genetic programming. But the ramified structures created by GEP (expression trees) are the expression of a totally autonomous genome [4].

There are two main players in gene expression programming: the chromosomes and the Expression Trees (ETs) or programs, being the latter the expression of the genetic information encoded in the former. As in nature, the process of information decoding is called translation and this translation implies obviously a kind of code and a set of rules. The genetic code of gene expression programming is very simple: a one-to-one relationship between the symbols of the chromosome and the nodes they represent in the trees. The rules are also very simple: they determine the spatial organization of nodes in the expression trees and the type of interaction between sub-ETs. Therefore, there are two languages in GEP: the language of the genes and the language of expression trees and, thanks to the simple rules that determine the structure of ETs and their interactions, it is possible to infer immediately the expression tree given the sequence of a gene, and vice versa. This unequivocal bilingual notation is

called Karva language. Figure 6 shows an example of expression trees and Karva language [3].

The fundamental steps of gene expression programming are schematically represented in Figure 7. The process begins with the random generation of the chromosomes of a certain number of individuals (the initial population). Then these chromosomes are expressed and the fitness of each individual is evaluated against a set of fitness cases (also called selection environment). The individuals are then selected according to their fitness (their performance in that particular environment) to reproduce with modification, leaving progeny with new traits. These new individuals are, in their turn, subjected to the same developmental process: expression of the genomes, confrontation of the selection environment, selection, and reproduction with modification. The process is repeated for a certain number of generations or until a good solution has been found [4].

The basis for the novelty of GEP resides on the revolutionary structure of GEP genes. The simple but plastic structure of these genes not only allows the encoding of any conceivable program but also allows their efficient evolution. Due to this versatile structural organization, a very powerful set of genetic operators can be easily implemented and used to search very efficiently the solution space. As in nature, the search operators of gene expression programming always produce valid structures and therefore are remarkably suited to creating genetic diversity [3].

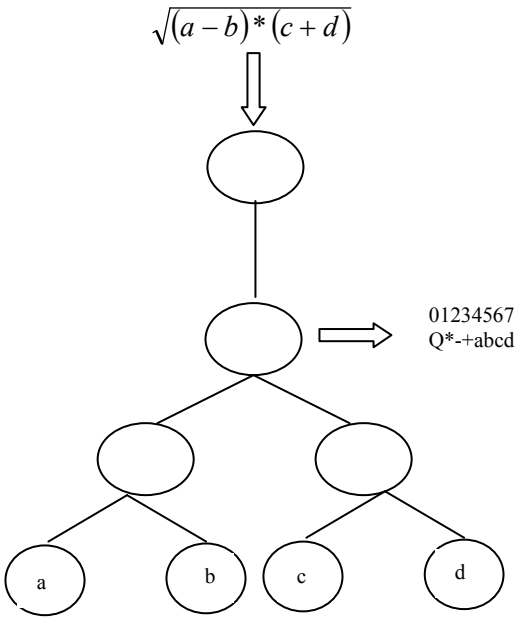

Figure 6. An example of expression trees and Karva language, adapted from [3].

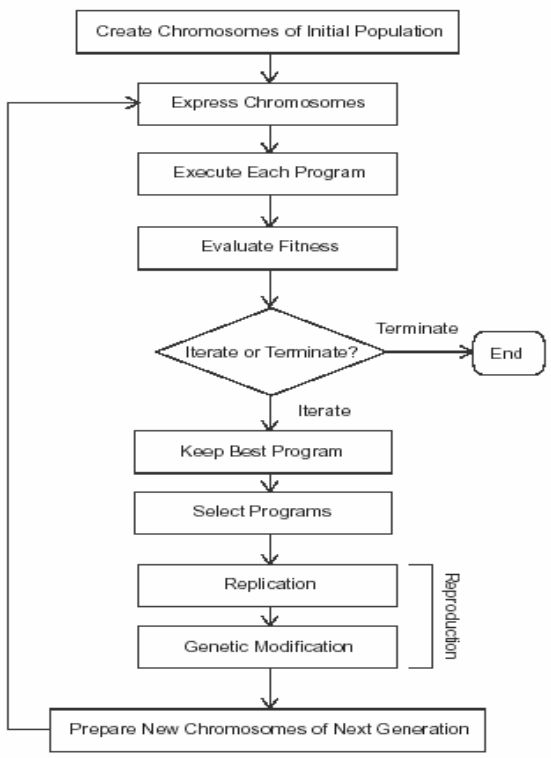

Figure 7. The flowchart of GEP, adapted from [4].

Automatic Problem Solver ® - APS 3.0 - (www.gepsoft.com), GEP software package, is used in modeling the hydraulic jump. About 65% of the collected experimental data (about 821 observations) are utilized to build the proposed GEP model in the light of Equation 3. The rest of the observations (442 observations) are used to test the built model.Table 1 shows the settings of the different genetic operators used in deriving the model while Table 2 shows the statistics obtained during the training and the testing phases of the model. APS 3.0 have the ability to automatically translates the models evolved in its native Karva code into a wide set of programming languages such as C, C++, C#, Visual Basic, VB.Net, Java, Java Script, and Fortran through the use of built-in grammars [3]. The Karva code of the obtained hydraulic jump model is shown in Figure 8 and the C# code of the same model is shown in Figure 9.

```
+.*.d0.-.IFA4.Acosh.d2.*.d0.d0.d2.d0.d1.d3.d2.d1.d2
+
Cos.IFA2.IFA1.Mod.d0.+.d1.IFA2.d4.d4.d3.d0.d3.d2.d0.d2.d0
+
Cos.IFA2.IFA1.Mod.d0.+.d1.+.d1.d1.d3.d0.d2.d0.d1.d3.d3
+
Sqrt.+.IFA4.Pow10.*.IFA2.Sin.Asinh.d4.d0.d1.d4.d4.d0.d3.d1.d1
+
IFA2.*.d0.-.Asinh.d0.IFA3.IFA4.d1.d4.d1.d0.d0.d4.d3.d0.d1
+
IFA4.d0./.-.Tanh.d0.*.Log.d2.d4.d0.d4.d0.d2.d2.d0.d2
```

Figure 8. Karva code of the hydraulic jump model.

Table 1. Genetic settings.

General	
Chromosomes:	30
Genes:	6
Head Size:	8
Gene Size:	17
Linking Function:	Addition
Genetic Operators	
Mutation Rate:	0.044
Inversion Rate:	0.1
IS Transposition Rate:	0.1
RIS Transposition Rate:	0.1
One-Point Recombination Rate:	0.3
Two-Point Recombination Rate:	0.3
Gene Recombination Rate:	0.1
Gene Transposition Rate:	0.1

Table 2. Statistics of the derived model.

	Training	Testing
Best Fitness:	132.097717883727	99.654689137223
Max. Fitness:	1000	1000
R-square:	0.938649917599482	0.914259359612276
Outliers:	--	--
Calc. Errors:	0	0
Correlation Coefficient (CC):	0.96883946946822	0.95616910617959
Mean Squared Error (MSE):	6.57015348955692	9.03465073904365
Root Mean Squared Error (RMSE):	2.56323106441010	3.00576957517433
Relative Absolute Error (RAE):	0.23522615900274	0.26305610654876
Mean Absolute Error (MAE):	1.81426189597787	2.04167307710747
Relative Squared Error (RSE):	0.06176698293869	0.08874924599674
Root Relative Squared Error (RRSE):	0.24852964197192	0.29790811670168

```
using System;
class apsModel
{
    public double Calculate(double[] d)
    {
        double dblTemp = 0.0;
        dblTemp = (((apsAcosh(d[0])-d[2])*((d[2]*d[0])>=d[0]?(d[2]*d[0]):d[0]))+d[0]);

        dblTemp += Math.Cos(((d[0]<(d[4]+d[4])?d[0]:(d[4]+d[4]))>Math.IEEERemainder
            (d[1],(d[3]>d[0]?d[3]:d[0]))?(d[0]<(d[4]+d[4])?d[0]:(d[4]+d[4])):
            Math.IEEERemainder(d[1],(d[3]>d[0]?d[3]:d[0]))));

        dblTemp += Math.Cos(((d[0]<(d[1]+d[1])?d[0]:(d[1]+d[1]))>Math.IEEERemainder
            (d[1],(d[3]+d[0]))?(d[0]<(d[1]+d[1])?d[0]:(d[1]+d[1])):Math.IEEERemainder
            (d[1],(d[3]+d[0]))));

        dblTemp += Math.Sqrt((((apsAsinh(d[4])*d[4])>=(d[0]>d[1]?d[0]:d[1])?
            (apsAsinh(d[4])*d[4]):(d[0]>d[1]?d[0]:d[1]))+Math.Pow(10,Math.Sin(d[4]))));

        dblTemp += (((d[0]-(d[1]<=d[4]?d[1]:d[4]))*apsAsinh((d[1]>=d[0]?d[1]:d[0])))>d[0]?
            ((d[0]-(d[1]<=d[4]?d[1]:d[4]))*apsAsinh((d[1]>=d[0]?d[1]:d[0]))):d[0]);

        dblTemp += (d[0]>=((d[0]-(d[2]*d[4]))/Math.Tanh(Math.Log10(d[0])))?d[0]:
            ((d[0]-(d[2]*d[4]))/Math.Tanh(Math.Log10(d[0]))));
        return dblTemp;
    }
    double apsAsinh(double x)
    {
        return Math.Log(x+Math.Sqrt(x*x+1));
    }
    double apsAcosh(double x)
    {
        return Math.Log(x+Math.Sqrt(x*x-1));
    }
}
```

Figure 9. The C# code of the hydraulic jump model.

6. Model Comparison

After MLR and GEP calculations were performed, results obtained from MLR and GEP, models were compared with measured values for both training and testing and validation data sets. Though various measures can be used for comparison purposes, for simplicity, only R-square and Root Mean Square Error (RMSE) are considered here. The performance of the MLR and GEP models are given in Table 3.

Table 3. The performance of the MLR and GEP models.

	Training Data		Testing and Validating Data	
	R^2	RMSE	R^2	RMSE
LJ/Y1-MLR	0.874	3.65	0.873	3.66
LJ/Y1-GEP	0.938	2.56	0.914	3.005

7. Conclusion

In the present study, about 1263 observations on hydraulic jumps formed in horizontal rectangular roughened bed are utilized to build a hydraulic jump model using two different modeling techniques. The first model was built using MLR and the second one is built using GEP. Two measures, R-square and RMSE, were used for comparing the performance of both models. The results indicate that GEP gives higher regression coefficient than MLR but the obtained GEP model is more complicated than the MLR model. It is concluded that GEP is a promising AI approach for hydraulic data modeling.

References

[1] Chow V., *Open Channel Hydraulics*, McGraw Hill, New York, 1981.

[2] Ferreira C., "Gene Expression Programming: A New Adaptive Algorithm for Solving Problems," *Complex Systems*, vol. 13, no. 2, pp. 87-129, 2001.

[3] Ferreira C., *Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence*, Angra do Heroismo, Portugal, 2002.

[4] Ferreira C., "Gene Expression Programming and the Evolution of Computer Programs," *in de Castro L. and Zuben F., Recent Developments in Biologically Inspired Computing*, pp. 82-103, 2004.

[5] Hager W., *Energy Dissipators and Hydraulic Jump*, Kluwer Academic Publishers, the Netherlands, 1992.

[6] Montgomery D. and Runger G., *Applied Statistics and Probability for Engineers*, John Wiley and Sons Inc, New York, 1999.

[7] Negm A., "Optimal Roughened Length of Prismatic Stilling Basins," *in Proceedings of the 5th International Conference on Hydro-Science and Engineering*, Poland, 2002.

[8] Negm A., Shouman M., and Abdel-Gawad A., "Performance Evaluation of Artificial Neural Networks Software in Prediction of Hydraulic Data," *in Proceedings of the 6th International Conference on Hydroinformatics*, pp. 1-8, Singapore, 2004.

[9] SPSS Inc., *Neural Connections® User's Guide*, USA, 1997.

[10] Vischer D. and Hager W., *Energy Dissipators*, *IAHR Hydraulic Structures Design Manual Series No.*, Balkema Publishers, Netherlands, 1995.

[11] Velickov S. and Solomatine D., "Predictive Data Mining: Practical Examples," *in Proceedings of the Second Joint Workshop on Artificial Intelligence in Civil Engineering*, pp. 1-17, Cottbus, Germany, 2000.

[12] Zhou C., Xiao W., Tirpak T., and Nelson P., "Discovery of Classification Rules by Using Gene Expression Programming," *in Proceedings of the 2002 International Conference on Artificial Intelligence*, pp. 1355-1361, Las Vegas, 2002.

Khalid Eldrandaly is an assistant professor of computer information systems and interim head of Information Systems and Technology Department, College of Computers and Informatics, Zagazig University, Egypt. He received his BSc degree in civil engineering, his MSc degree in systems engineering, and his PhD degree in systems engineering (GIS). He was a visiting scholar at Texas A&M University, USA, for two years. His research interests include GIS, expert systems, SDSS, MCDM, and intelligent techniques in decision making. He is a member of the World Academy of Young Scientists (WAY), Arab Union of Scientists and Researchers (AUSR), Texas A&M International Faculty Network, and Egyptian Software Engineers Association (ESEA).

Abdel-Azim Negm is a full professor of hydraulic engineering, College of Engineering, Zagazig University, Egypt. He received his BSc degree in irrigation and environmental engineering, and his MSc and PhD degrees in hydraulic engineering. He supervised more than ten MSc's and PhD's thesis in the field of hydraulic engineering and its applications. Dr. Negm published more than 140 scientific papers in the feild of hydraulic engineering in the national and international journals and conferences. About ten papers were related to ANN and expert systems applications in hydraulic engineering.