American-Eurasian J. Agric. & Environ. Sci., 4 (5): 575-583, 2008
ISSN 1818-6769

On the Applicability of Linear Genetic Programming for the Formulation of Soil Classification

[1]A.A.R. Heshmati, [1]H. Salehzade, [1]A.H. Alavi, [1,2]A.H. Gandomi, [3]A. Badkobeh and [1]A. Ghasemi

[1]College of Civil Engineering, Iran University of Science and Technology, Tehran, Iran
[2]College of Civil Engineering, Tafresh University, Tafresh, Iran
[3]College of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran

Abstract: The main purpose of the present study is to propose new formulations for soil classification by means of a promising variant of genetic programming (GP) namely, linear genetic programming (LGP) for the first time in the literature. Properties of soil namely, plastic limit, liquid limit, color of soil, percentage of gravel, sand and fine grained particles are used as input variables to the models to predict the classification of the soils. The models are developed using a reliable database obtained from the previously published literature. The results of proposed formulations are further compared with the existing models found in the literature. The results demonstrate that the LGP based formulas are able to predict the target values to high degree of accuracy.

Key words: Soil classification · Linear genetic programming · Formulation

INTRODUCTION

The main objectives of soil classification are to investigate the effect of classification on the fertility of soil when it is irrigated [1] and to find the suitability of the soil for the construction of different structures like dams and embankments subgrade [2]. A range of complex factors affect the naming of soils because the soils are not usually available in nature separately as sand, gravel or any other single component but mostly are found as mixture with varying proportions of particles of different size [3]. For instance, sandy clay has most of the properties of clay but contains a significant amount of sand. While the behavior of soil mass under load deeply depends on the various constituents present in the mass, the degree of density saturation and environmental conditions, the soil is given the name of the constituent that appears to have significant influence on its behavior. Therefore, it is idealistic to develop predictive models to able to evaluate the classification of the soil and overcome the limitations of existing classification systems by considering all factors related to the soil formation.

Genetic programming (GP) [4, 5] is a developing subarea of evolutionary algorithms [6] inspired from Darwin's evolution theory. GP may be defined generally as a supervised machine learning technique that searches a program space instead of a data space [5]. In recent years, a particular subset of GP with a linear structure similar to the DNA molecule in biological genomes namely, linear genetic programming (LGP) [7] has been emerged. LGP is a machine learning approach that evolves the programs of an imperative language or machine language instead of the traditional Koza's tree-based GP [4] expressions of a functional programming language. Despite significant advantages of LGP over the other modelling approaches, there has been just some little scientific effort directed at applying it to real-world data sets, [7-9].

The main purpose of this paper is to utilize LGP technique to obtain formulas for the determination of soil classification. To our knowledge, this is the first time in the literature to utilize this approach to introduce explicit formulations of the soil classification. A comparison between the results of proposed formulas, as well as existing models found in the literature, was conducted in terms of prediction quality. A reliable database including previously published soil classification test results was utilized to develop the models.

Review of Previous Studies: A few studies have concentrated on assessing the classification of soils by using artificial neural networks (ANN) [10, 11] in the

Coresponding Author: Dr. Ali Akbar Heshmati R., College of Civil Engineering, Iran University of Science and Technology, Tehran, Iran

literature [12-15]. Despite the successful performance of ANNs, they are black-box models that usually do not give a deep insight into the process they use the available information to obtain a solution and also a better understanding of the nature of the derived relationship between the different interrelated input and output data is not provided.

There has been only limited research with the specific objective of opening ANN models adequately and introducing explicit formulations of soil classification by means of them [15] built an empirical model using a sequential learning approach (SLA) for single hidden radial basis function (RBF) neuron neural networks proposed by [16]. Their developed sequential learning neural network (SLNN) model was utilized for the prediction of soil classification. The values for learning rate and gamma have been respectively chosen as 0.6 and 1e–12 for the architecture of their proposed SLNN (RBF) model. In that work, an equations was introduced based on experimental results and by using the values of the weights obtained from neural network training to predict the soil classification (*SC*) that is given as follows:

$$SC_{SLNN} = 2653.92e^{-W_1} \tag{1}$$

where,

$$W_1 = \frac{1}{(0.80439)^2}[(x_1 - 1.1843)^2 + (x_2 + 1.0463)^2 + (x_3 - 0.8604)^2 + (x_4 - 1.0218)^2 + (x_5 + 0.2890)^2 + (x_6 - 2.3476)^2 \tag{2}$$

where

x_1: Color of soil

x_2: Percentage of gravel

x_3: Percentage of sand

x_4: Percentage of fine grained particles

x_5: Percentage of liquid limit

x_6: Percentage of plastic Limit

and x_1,..., x_6 are the six input parameters to the model. For inputs to the SLNN network the following rule was used for the color of the soil. 0.1:Brown; 0.2:Brownish Grey; 0.3:Grayish Brown; 0.5:Reddish yellow; 0.7:Yellowish Red.

The output of the network is the classification of the soil which is given as: 0.1: Clayey soil; 0.2: Clay with medium compressibility; 0.3: Clay of low compressibility; 0.6: Silt with medium compressibility.

It should be noted that the required data used for the training and testing of the SLNN model described above were taken from [12] and have been also utilized in the present study.

Genetic Programming (GP): Genetic programming (GP) is one of the branches of evolutionary methods that creates computer programs to solve a problem using the principle of Darwinian natural selection. GP was introduced by Koza as an extension of the genetic algorithms, in which programs are represented as tree structures and expressed in the functional programming language LISP [4]. A comprehensive description of GP is beyond the scope of this paper and can be found in [4, 5]. GP has been successfully applied to some of the civil engineering problems [17-21].

Linear Genetic Programming (LGP): Linear genetic programming (LGP) is a subset of GP that has been emerged recently. Comparing LGP to the traditional Koza's tree-based GP, there are some main differences such as the graph-based functional structure of linear genetic programs (LGPs), evolvement of these programs in an imperative programming language (like c/c ++) [22] and machine code [23] rather than in expressions of a functional programming language (like LISP) and the coexistence of structurally noneffective code with effective code in LGPs.

Noneffective code in genetic programs which is referred to as "intron", represents instructions without any influence on the program behavior. Structural introns act as a protection that reduces the effect of variation on the effective code and also allow variations to remain neutral in terms of fitness change. Because of the imperative program structure in LGP, these noneffective instructions can be identified efficiently. This allows the corresponding effective instructions to be extracted from a program during runtime. Since, only these effective programs are executed when testing fitness cases, evaluation can be accelerated significantly (Fig. 1).

The instructions from imperative languages are restricted to operations that accept a minimum number of constants or memory variables, called registers (r) and assign the result to a destination register, e.g., r0:= r1 + 1. A part of a linear genetic program in C code is represented as follows [7]:

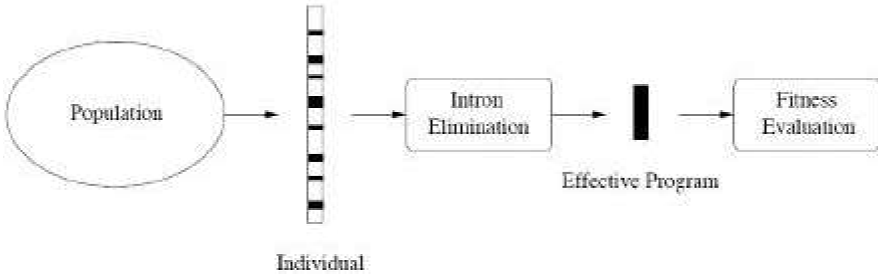

Fig. 1: Elimination of noneffective code in LGP. Only effective program is executed [7]

```
void LGP (double r[5])
{...
r[0] = r[5] + 70;
r[7] = r[0] - 50;
if (r[1] > 0)
if (r[5] > 2)
r[4] = r[2] * r[1];
r[2] = r[5] + r[4];
r[6] = r[4] * 10;
r[1] = r[3] / 2;
if (r[0] > r[1])
r[3] = r[5] * r[5];
r[7] = r[6] - 2;
if (r[1] <= r[6])
r[0] = sin(r[7]);
}
```

where register r[0] holds the final program output. LGPs can be converted into a functional representation by successive replacements of variables starting with the last effective instruction [24]. Automatic Induction of Machine code by Genetic Programming (AIMGP) is a particular form of LGP. AIMGP induces binary machine code directly without any interpreting steps that results in a significant speedup in execution compared to interpreting GP systems. This LGP approach searches for the computer program and the constants at the same time. The evolved program is a sequence of binary machine instructions [23].

The machine-code-based, LGP uses the following steps to evolve a computer program that predicts the target output from a data file of inputs and outputs [7, 25]:

Step1: Initializing a population of randomly generated programs.

Step2: Running a tournament. In this step four programs are selected from the population randomly. They are compared and based on fitness two programs are picked as the winners and two as the losers.

Step3: Transforming the winner programs. After that two winner programs are copied and transformed probabilistically as follows:

- Parts of the winner programs are exchanged with each other to create two new programs (crossover operation); and/or
- Each of the tournament winners are changed randomly to create two new programs (mutation operation).

Step4: Replacing the loser programs in the tournament with the transformed winner programs. The winners of the tournament remain without change.

Step5: Repeating steps two to four until convergence. A program defines the output of the algorithm that simulates the behavior of the problem to an arbitrary degree of accuracy.

Detail of LGP and a comprehensive description on basic LGP parameters can be obtained from [7].

Model Development: The details of developing the LGP based models including the database description and comparison of the performance of the models are presented in the following subsections.

Database: In the present study, out of different classification systems of soils, the unified soil classification or IS classification system is considered. In practice, soil classification is determined comparing a data with the existing experimental results. The Bureau of Indian Standards classifies soils based on the color of soil (CS), percentages of gravel (%G), sand (%S), fine grained particles (%F), liquid limit (LL) and plastic Limit (PL). These six important properties are utilized as the input parameters to the LGP models to predict the soil

Table 1: Database used in developing the models

Sample No.	CS	G (%)	S (%)	F (%)	LL (%)	PL (%)	SCTest
Training							
1	0.1	0	80	20	36	22	0.1
2	0.2	2	56	42	30	18	0.1
3	0.1	0	28	72	43	26	0.2
4	0.1	0	27	73	42	25	0.2
5	0.2	0	43	57	34	23	0.3
6	0.2	0	42	58	34	22	0.3
7	0.2	0	45	55	35	23	0.3
8	0.2	3	55	42	31	19	0.1
9	0.2	0	48	52	36	28	0.6
Testing							
10	0.2	4	56	40	30	18	0.1
11	0.1	0	28	72	42	25	0.2
12	0.2	0	44	56	34	22	0.3
13	0.2	0	42	58	35	23	0.3
14	0.1	0	76	24	37	23	0.1
15	0.2	4	54	42	31	18	0.1
16	0.1	0	78	22	36	23	0.1
17	0.1	0	28	72	43	25	0.2

Table 2: The variables used in model development

Parameters	Range	Normalization value	Code
Inputs			
Color of soil (CS)	0.1-0.2	-	x1
Gravel (%G)	0-4	18	x2
Sand (%S)	27-80	82	x3
Fine grained particles (%F)	20-72	84	x4
Liquid limit (%LL)	30-43	59	x5
Plastic Limit (%PL)	18-28	34	x6
Output			
Soil classification (SC)	0.1-0.6	-	-

classification (SC) as the single output. The following rule was used for the color of the soil as an input to the LGP models.

0.1: Brown;
0.2: Brownish grey;
0.3: Grayish brown;
0.5: Reddish yellow;
0.7: Yellowish red.

Similar to SLNN network, the output of LGP is the classification of the soil which is given as:

0.1: Clayey soil (SL)
0.2: Clay with medium compressibility (CI)
0.3: Clay of low compressibility (CL)
0.6: Silt with medium compressibility (MI)

Table 3: Parameter settings for LGP

Parameter	Settings SC1	Settings SC2
Function set	+, -, *, /, √, sin, cos, tan	+, -, *, /
Population size	2000-5000	2000-5000
Maximum program size	256	256
Initial program size	80	80
Crossover rate (%)	50, 95	50, 95
Homologous crossover (%)	95	95
Mutation rate %)	90	90
Block mutation rate (%)	30	30
Instruction mutation rate (%)	30	30
Data mutation rate (%)	40	40
Number of demes	20	20

The database for model construction contains 17 soil classification test results reported by [12]. Table 1 shows the experimental database used for the development of the models. The input and output parameters entering the models have been normalised between 0 and 1 before the learning process. The range of samples, normalization values and the format of the input data used in this study are given in Table 2.

Model Development Using LGP: The main goal is to obtain the explicit formulations of soil classification as a function of variables given as follows:

$$SC = f\left(CS, G, S, F, LL, PL\right) \qquad (3)$$

The six parameters are used for the LGP models as the input variables. Two LGP models for single output have been separately developed in order to obtain two different formulas for soil classification by using different function sets for runs. The first function set consists of nearly all functions and the latter includes just addition, subtraction, division and multiplication in order to obtain short and very simple formulas. The various LGP involved parameters are population size, mutation rate and its different types (block mutation rate, instruction mutation rate and data mutation rate), crossover rate and homologous crossover, function set, number of demes and program size. The parameter selection will affect the model generalization capability of LGP. They were selected based on some previously suggested values [26, 27] and also after a trial and error approach. The parameter settings have been shown in Table 3.

For the LGP based models, Discipulus software [26] is used which is based on the AIMGP approach. For the analysis, the data sets are divided into training and testing subsets. Out of the 17 data sets, presented in

Table 1, the first 9 values of are taken for training the LGP algorithm and the next 8 values were used for testing the generalization capability of the models. In order to evaluate the capabilities of the proposed LGP models, the correlation coefficient (R), mean squared error (MSE) and mean absolute error (MAE) are used as the criteria between the actual and predicted values. R, MSE and MAE are calculated using the following equations:

$$R = \frac{\sum_{i=1}^{n}\left(h_i - \overline{h}_i\right)\left(t_i - \overline{t}_i\right)}{\sqrt{\sum_{i=1}^{n}\left(h_i - \overline{h}_i\right)^2 \sum_{i=1}^{n}\left(t_i - \overline{t}_i\right)^2}} \tag{4}$$

$$MSE = \left[\frac{\sum_{i=1}^{n}\left(h_i - t_i\right)^2}{n}\right] \times 100 \tag{5}$$

$$MAE = \left[\frac{1}{n}\sum_{i=1}^{n}\left|h_i - t_i\right|\right] \times 100 \tag{6}$$

where h_i and t_i are respectively the actual output and the calculated output value for the ith output, $\overline{h}_i$ is the average of the actual outputs,and n is the number of sample. The other details of soil classification predictive models are highlighted in next sections.

Explicit Formulation of Soil Classification and Analysis Using LGP Models: Formulation of soil classification in functional form in terms of the independent variables, color of soil (CS = x_1), gravel (%G = x_2), sand (%S = x_3), fine grained particles (%F = x_4), liquid limit (LL = x_5) and plastic Limit (PL = x_6) for the best test R values by the LGP algorithm are given in Eq. (7) and Eq. (8) for two different function sets.

$$SC_1 = x_1x_6^3(18(\sin(\sin(\sin(\sin(14(18x_1x_6(\sin(\sin(x_1 + \frac{x_2}{18} + \frac{59x_6}{17x_1x_5})))^2$$
$$+2x_1x_6(9x_1 - x_2) - 17(18x_1 + x_2))/51x_4 + x_1)) - \frac{x_2}{18})) + x_2 - 54)^2)$$
$$/25468992 \tag{7}$$

$$SC_2 = \frac{x_1x_6}{34}(\frac{x_6}{34}(\frac{2x_6}{17}(\frac{x_1x_6}{68}(-\frac{x_1x_5}{59} - \frac{2x_2}{9} + \frac{2x_6}{17}) - \frac{x_5}{236} - \frac{x_2}{36} + \frac{x_6}{68}) -$$
$$\frac{2x_5}{59} - \frac{2x_2}{9} + \frac{2x_6}{17}) + \frac{x_4}{84}) \tag{8}$$

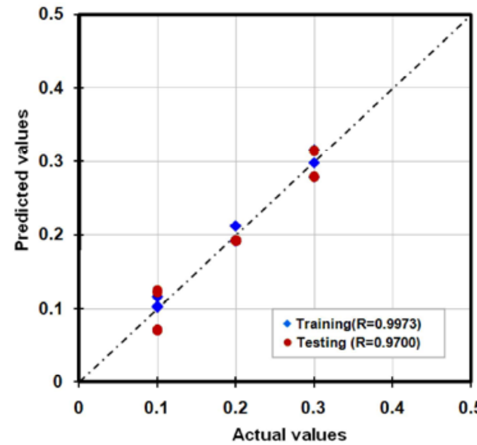

Fig. 2: Results of LGP prediction and actual soil classification obtained by Eq. (7)

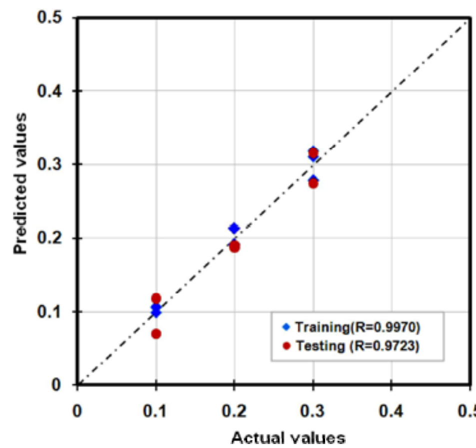

Fig. 3: Results of LGP prediction and actual soil classification obtained by Eq. (8)

The comparison of LGP prediction and actual soil classification for Eq. (7) is shown in Fig. 2. It can be seen from this figure that Eq. (7) generated by LGP model yielded high R values equal to 0.9973 and 0.97 for training and testing data, respectively. Fig. 3 shows the relevant results obtained by Eq. (8). It can be observed from this figure that Eq. (8) yielded R values equal to 0.9970 and 0.9723 for training and testing data, respectively.

In order to evaluate how many times each input appears in a way that contributes to the fitness of the LGP programs that contain them (importance of input parameters), frequency [28] values of input parameters of the strength soil classification predictive models are obtained and presented in Fig. 4. A value of 1.00 in this figure indicates that this input variable appeared in 100% of the best thirty programs evolved by LGP. To obtain the

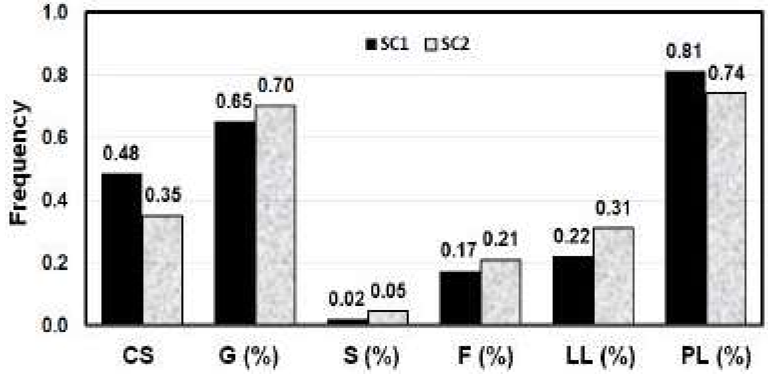

Fig. 4: Frequency values of input parameters of soil classification predictive models

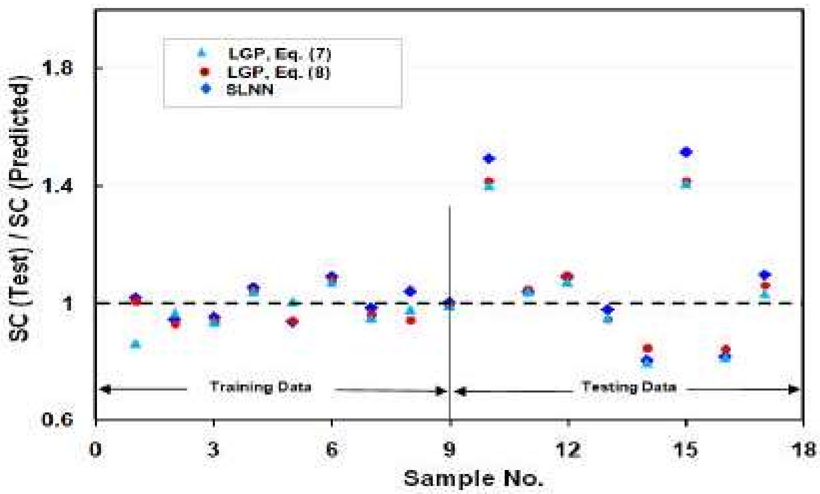

Fig. 5: Soil classification relative comparison for all element tests data

frequency values, the data sets were divided into training (training and validation) and testing subsets as described above. The analysis was performed by using the version 3 of Discipulus software [28]. The frequency values, presented in Fig. 4, are achieved for the best test R values of LGP runs. Fig. 4 shows that the frequencies of CS, %G, %S, %F, LL and PL are respectively equal to 0.48, 0.65, 0.02, 0.17, 0.22 and 0.81 for the first model (SC_1). The relevant results for the second model (SC_2) demonstrate that the frequencies of CS, %G, %S, %F, LL and PL are equal to 0.35, 0.7, 0.05, 0.21, 0. 31 and 0.74, respectively. According to these results, it can be found that for both of the proposed models, soil classification is more sensitive to %G and PL in comparison with the other inputs. Moreover, the functions generated for the best result by the LGP models consider the effects of all parameters except x_3 (%S). According to frequency values, classification of soil is less sensitive to %*S* compared with the other inputs and it may be the reason of why it has not been directly incorporated in the LGP based formulas.

RESULTS AND DISCUSSION

In the present study, two formulas for the classification of soil in functional form in terms of CS, %G, %S, %F, LL and PL were obtained by using LGP and given in Eq. (7) and Eq. (8). As mentioned previously, R, MSE and MAE are selected as the target statistical parameters to evaluate the performance of the models. Fig. 5 represents the results for all element test data. Statistical performance of LGP based formulations, as well as SLNN model [15], in terms of their prediction capabilities are summarized in Table 4. Comparing the performance of the LGP based formulas, it can be observed that the best performance is achieved by Eq. (7) on the training data (R = 0.9973, MSE=0.0017,

Table 4: Statistical performance of models for soil classification prediction

	Training			Testing			All elements		
Models	R	MSE	MAE	R	MSE	MAE	R	MSE	MAE
LGP, Eq. (7)	0.9973	0.0017	0.1052	0.9700	0.0044	0.1842	0.9916	0.0047	0.2444
LGP, Eq. (8)	0.9970	0.002	0.1141	0.9723	0.0062	0.2081	0.9916	0.0054	1.561
SLNN	0.9967	0.0013	0.0871	0.9657	0.0086	0.2337	0.9904	0.0103	1.906

Table 5: Comparative analysis of proposed LGP based formulae with experimental and SLNN results

Sample No.	SC_{Test}	SC_1, Eq. (7)	SC_{Test}/SC_1	SC_2, Eq. (8)	SC_{Test}/SC_2	SC_{SLNN}	SC_{Test}/SC_{SLNN}
Training							
1	0.1	0.116	0.86	0.099	1.01	0.098	1.02
2	0.1	0.103	0.97	0.107	0.93	0.106	0.94
3	0.2	0.213	0.94	0.213	0.94	0.210	0.95
4	0.2	0.192	1.04	0.192	1.04	0.190	1.05
5	0.3	0.298	1.01	0.319	0.94	0.320	0.94
6	0.3	0.279	1.07	0.278	1.08	0.275	1.09
7	0.3	0.315	0.95	0.311	0.96	0.305	0.98
8	0.1	0.102	0.98	0.106	0.95	0.096	1.04
9	0.6	0.604	0.99	0.601	1.00	0.597	1.01
Testing							
10	0.1	0.071	1.40	0.071	1.42	0.067	1.49
11	0.2	0.192	1.04	0.191	1.05	0.192	1.04
12	0.3	0.279	1.07	0.275	1.09	0.275	1.09
13	0.3	0.315	0.95	0.316	0.95	0.306	0.98
14	0.1	0.125	0.80	0.118	0.85	0.124	0.81
15	0.1	0.071	1.41	0.071	1.42	0.066	1.52
16	0.1	0.122	0.82	0.119	0.84	0.122	0.82
17	0.2	0.193	1.04	0.188	1.06	0.182	1.10

MAE = 0.1052). Considering the testing data, it can be seen that Eq. (8) with R, MSE and MAE values equal to 0.9723, 0.0062 and 0.2081 outperforms Eq. (7).

Comparing the results of the LGP based formulas and those of SLNN, it can be seen that both of the formulae obtained by LGP approach perform superior than the SLNN on training, testing and all element tests data. The results for all element tests data also demonstrate that while both of Eq. (7) and Eq. (8) yielded R values equal to 0.9916, the former outperforms the latter regarding its lower MSE and MAE values. Table 5 shows a comparative analysis of results of the proposed LGP formulations and the results obtained by SLNN model including soil classification actual experimental values.

SUMMARY AND CONCLUSIONS

In this paper, the first application of a particular subset of GP namely, LGP to the soil classification prediction is presented along with its performance comparisons. Two formulas for the classification of soil have been obtained by means of LGP and considering two different function sets. A reliable database including previously published soil classification test results was used for training and testing the prediction models. The LGP based formulations results were compared with the experimental results and the existing model proposed in the literature namely, SLNN (RBF).

The values of performance measures for the models indicate that the proposed LGP models are able to predict the target values to high degree of accuracy. The results also demonstrate that for the prediction of soil classification both of the formulae evolved by LGP outperform the results of SLNN model.

In addition to the considerable accuracy of LGP based prediction equations, they are quite short and very simple and seem to be more practical for use compared to the equations produced by SLNN. However, this investigation revealed that LGP is very promising approach that can be utilized in order to capture the underlying relationship between the different interrelated input and output data for many of civil engineering tasks.

REFERENCES

1. Keshavarzpour, F. and M. Rashidi, 2008. Effect of Different Tillage Methods on Soil Physical Properties and Corp Yield of Watermelon. World Appl. Sci. J., 3(3): 359-364.
2. Rashidi, M. and K. Seyfi, 2008. Comparative studies on bekker and upadhyaya models for soil pressure-sinkage behaviour prediction. American-Eurasian J. Agric. and Environ. Sci., 3(1): 19-25.
3. Mostashari, M., M. Muazardalan, N. Karimian, H.M. Hosseini and H. Rezai, 2008. Phosphorus Fractions of Selected Calcareous Soils of Qazvin Province and their Relationships with Soil Characteristics. American-Eurasian J. Agric. and Environ. Sci., 3(4): 547-553.
4. Koza, J.R., 1992. Genetic programming: On the programming of computers by means of natural selection. Cambridge (MA): MIT Press.
5. Banzhaf, W., P. Nordin, R. Keller and F. Francone, 1998. Genetic Programming – An Introduction: On the Automatic Evolution of Computer Programs and Its Application. Heidelberg/San Francisco: Morgan Kaufmann.
6. Bäck, T., 1996. Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. USA: Oxford University Press.
7. Brameier, M. and W. Banzhaf, 2007. Linear Genetic Programming. New York (NY): Springer Science + Business Media.
8. Baykasoğlu, A., H. Güllü, H. Çanakçı and L. Özbakır, 2007. Prediction of compressive and tensile strength of limestone via genetic programming. Expert Systems with Applications, Article in press.
9. Deschaine, LM., 2000. Using Genetic Programming to Develop a C/C++ Simulation Model of a Waste Incinerator. Science Applications International Corp., Draft Technical Report.
10. Haykins, S., 1999. Neural networks – A comprehensive foundation (2nd Ed.). Englewood Cliffs (NJ): Prentice Hall International Inc.
11. Shayanfar, M.A., S.R. Massah and H. Rahami, 2007. An Inverse Reliability Method Using Neural Networks and Genetic Algorithms. World Appl. Sci. J., 2(6): 594-601.
12. Suresh, D., 2000. Application of neural networks to civil engineering problems. M.E. Thesis, Bharathiar University, Coimbatore, India.
13. Cal, Y., 1995. Soil Classification by neural network. Advances in Engineering Software, 22(2): 95-97.
14. Levine, E.R., D.S. Kimes and V.G. Sillito, 1996. Classifying soil structure using neural networks. Ecological modeling, 92(1): 101-108
15. Rajasekaran, S. and R. Amalraj, 2002. Predictions of design parameters in civil engineering problems using SLNN with a single hidden RBF neuron. Comput and Struc, 80(31): 2495-505.
16. Zhang, J. and A.J. Morris, 1988. A sequential learning approach for single hidden layer neural networks. Neural Networks, 11(1): 65-80.
17. Heshmati, A.A., A.H. Alavi, AH. Gandomi and M. Gandomi, 2008. A Multi Expression Programming Application to High Performance Concrete. World Appl. Sci. J., submitted for publication.
18. Ashour, A.F., L.F. Alvarez and V.V. Toropov, 2003. Empirical modelling of shear strength of RC deep beams by genetic programming. Comput and Struc, 81(55): 331-338.
19. Alavi, A.H., A.H. Gandomi, M. Gandomi, MG. Sahab, 2008. A data mining approach to compressive strength of CFRP Confined Concrete Cylinders. Comput and Struc, submitted for publication.
20. Alavi, A.H., A.A.R. Heshmati, A.H. Gandomi, A. Askarinejad and M. Mirjalili, 2008. Utilisation of Computational Intelligence Techniques for Stabilised Soil. In: M. Papadrakakis and BHV. Topping editors. Proceedings of the 6th International Conference on Engineering Computational Technology, Civil-Comp Press, Stirlingshire, United Kingdom, paper, 175.
21. Gandomi, A.H., A.H. Alavi, S. Kazemi and M.M. Alinia, 2008. Behavior appraisal of steel semi-rigid joints using Linear Genetic Programming. J. Const. Steel Res., submitted for publication.
22. Brameier, M., W. Kantschik, P. Dittrich and W. Banzhaf, 1998. SYSGP – A C++ library of different GP variants. Technical Report [CI-98/48], Collaborative Research Center 531, University of Dortmund, Germany.
23. Nordin, P.J., 1994. A compiling genetic programming system that directly manipulates the machine code (Chapter 14). In: Kenneth E. and Jr. Kinnear editors. Advances in Genetic Programming, MIT Press, USA, pp: 311-31.

24. Oltean, M. and C. Grosan, 2003. A comparison of several linear genetic programming techniques. Complex Sys, 14(4): 1-29.
25. Deschaine, L.M., 2000. Tackling real-world environmental challenges with linear genetic programming. PCAI magazine, 15(5): 35-37.
26. Francone, F., 2004. Discipulus Lite™ Owner's Manual, Version 4.0. Register Machine Learning Technologies.
27. Feldt, R. and P. Nordin, 2000. Using Factorial Experiments to Evaluate the Effect of Genetic Programming Parameters. In: EuroGP 2000, LNCS 1802, pp: 271-82.
28. Francone F., 2000. Discipulus™ Owner's Manual, Version 3.0. Register Machine Learning Technologies.