Send Orders for Reprints to reprints@benthamscience.net

78 *The Open Civil Engineering Journal,* **2014**, *8*, 78-104

Open Access

Deep Pile Foundation Settlement Prediction Using Neurofuzzy Networks

Hussein Y. Aziz*

College of Engineering, Muthanna University, Sammawa, Muthanna, Iraq

Abstract: A NeuroFuzzy System (NFS) is one of the most commonly used systems in the real life problems because it has explicit and transparency which results from the fuzzy systems, with the learning and generalization capabilities from the dynamic behavior of the neural networks. It is one of the most successful systems, which introduced to decrement the fuzzy rules that constituting the underlying model. This system has a high efficiency; it gives good results in high speed. The NFS used in this study to predict the settlement of deep pile foundations. The results obtained from this system give good agreement and high precious for prediction of settlement compared with hyperbolic model and statistical regression analysis. Also, this scenario can be applied for similar or more complicated problems in the geotechnical engineering.

Keyword: Hyperbolic model, neurofuzzy system, pile foundation, settlement monitoring, statistical analysis.

1. INTRODUCTION

The design of foundations is generally controlled by the criteria of bearing capacity and settlement; the latter is often governing. The problem of estimating the settlement of foundations is very complex, uncertain and not yet entirely understood. A developed neural network was used for the prediction of settlement of a vertically loaded pile foundation in a homogeneous soil stratum [1]. The input variables used in this study for the proposed neural network consisted of the time and load according to the field measurements. The desired output from the Neurofuzzy (NF) model training is obtained by means of field measurements results of the settlement. The NF output is the pile foundation settlement.

Over the last few years, the use of NF has increased in many areas of geotechnical engineering. The use of NF networks in which, the acquired knowledge can be translated into a set of fuzzy rules that describe the relationship between the network inputs and the corresponding outputs in a transparent fashion. In the present study, the ability of NF networks to predict settlement of deep pile foundations in deep soft soils and to assist with providing a better understanding regarding the relationships between settlement and the factors affecting on settlement is assessed. In the last years, when the fuzzy systems are being considered as highly nonlinear or they includes a large number of input variables, the number of fuzzy rules constituting the underlying model is usually large. Dealing with a large size fuzzy model can pose many practical problems, such as, the increase of training time for the system's weights and the difficulty of updating them [2-4]. However, the model used in this study deals with the NF network to overcome these difficulties of the practical problems representing the settlement behavior in the deep soft soil area.

*Address correspondence to this author at the College of Engineering, Muthanna University, Sammawa, Muthanna, Iraq; Tel: 009647819731727; E-mail: husseinyousif_9@yahoo.com; husseinyousifaziz@gmail.com

2. NEUROFUZZY NETWORKS

2.1. Neurofuzzy Systems (NFS)

Existing fuzzy reasoning techniques suffer from the lack of a definite method to determine membership functions and a learning capability which can be overcome by neural networks driven fuzzy reasoning. Neural networks are used to tune the membership functions of fuzzy systems that are employed for controlling equipment. Although fuzzy logic has the ability to convert expert knowledge directly into fuzzy rules, it usually takes a lot of time to design and adjusts the linguistic labels (fuzzy sets) of the problem. In addition, the tuning of membership functions is a tricky procedure as it sometimes embodies a number of free parameters that must be assigned by an expert [3, 4]. Neural network techniques can automate this design procedure improving the performance and reducing the computational time. Neural network approach to design membership functions was proposed in 1989 [3]. The parameters, which define the shape of the membership functions, are modified to reduce error between output of the fuzzy system and supervised data [5-7].

The resultant combined system for the Fuzzy System and the Neural Network is called NFS, which posses the advantage of both, and overcomes some of the drawbacks of individual approaches, such as black-box of neural networks and the limited learning capability of fuzzy systems [3, 5]. One additional advantage of NF networks is that available engineering knowledge can be incorporated into the trained network to optimize model performance and to enhance the interpretation of a constructed model [8].

2.2. NeuroFuzzy Network Structure

The structure of NFS is determined by the functions used to represent the fuzzy sets. A general layout of NF network, which used Mamdani fuzzy model, with multi-inputs and one output, is shown in Fig. (**1**). The architecture of this

1874-1495/14 2014 Bentham Open

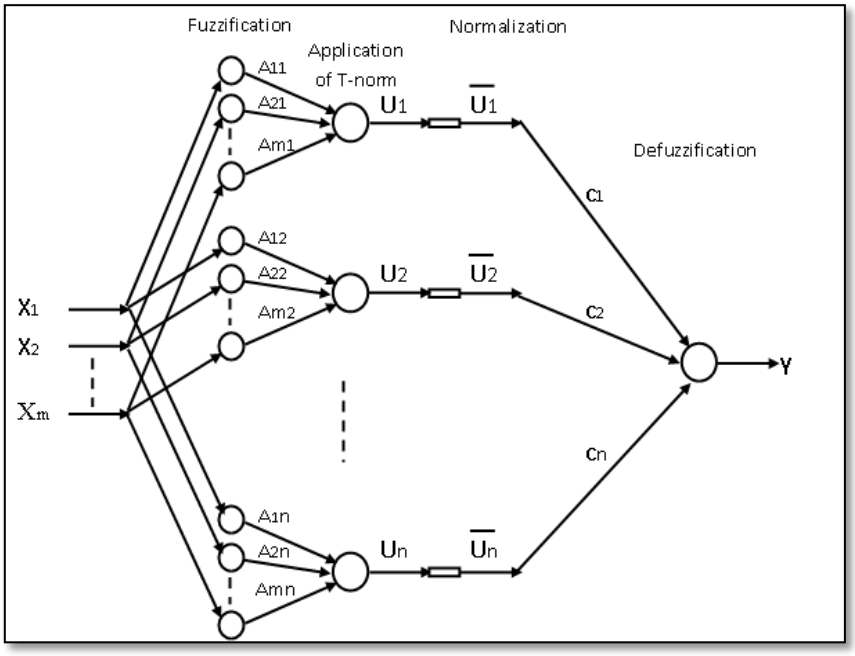


Fig. (1). NeuroFuzzy system structure.

network is analogous to that of artificial neural network with four-layers [3, 4, 7].

In the first layer, the fuzzification operator for each linguistic variable is performed using Gaussian membership functions, which is given in the following:

$$A_{ij}\left(x_j\right) = e^{-\frac{1}{2}\left(\frac{x_j - a_{ij}}{b_{ij}}\right)^2} \tag{1}$$

Where x_j is the j^{th} input variable; a_{ij} and b_{ij} are the center and the width for the Gaussian membership function.

The outputs of the first layer are fed to the next layer that performs a T-norm operation (product operation) [3]. The output of this layer represents the firing strength of premise (antecedent) part for each rule, which can be calculated as following:

$$U_i = \prod_{j=1}^{m} A_{ij} \tag{2}$$

Where: m is the number of input variables.

The firing strength is normalized in the third layer through dividing its value by the summation of all the firing strengths of all rules as:

$$U_i = \frac{U_i}{\sum_{L=1}^{n} U_L} \tag{3}$$

Where: n is the number of rules.

Finally, in the fourth layer, the summation of all the normalized values of U_i are multiplied by the corresponding weight c_i which represents the center of membership function in the consequent-part of the rules [2-4], to produce the center-of-gravity defuzzification operation. The output of the fourth layer represents the crisp output value for the given inputs, which can be obtained by the following formula:

$$Y = \frac{\sum_{i=1}^{n} U_i \, c_i}{\sum_{i=1}^{n} U_i} \tag{4}$$

2.3. NeuroFuzzy System and Supervised Learning

The adjusted parameters in the NF network can be divided into two categories based on the IF-part (premise-part) and THEN-part (consequent-part) parameters of the fuzzy rules [3, 5, 7]. In the premise-part, the center a_{ij} and the width b_{ij} of the Gaussian membership functions are being fine-tuned, whereas in the consequent-part, the adjusted parameters are the consequent weight c_i.

A gradient descent based back propagation algorithm is used LMS error-function to adjust the parameters of the NF network using the training patterns [3]. The main goal of supervised learning algorithm is to minimize this error-function, which has the formulas:

$$E^p(Z) = \tfrac{1}{2}\left(\mathrm{Yd}^{\mathrm{p}}(t) - Y^p(t)\right)^2 \tag{5}$$

$$E(Z) = \sum_{p=1}^{P} E^p(Z) \tag{6}$$

Where: E is the total error for the NFS.

E^p is the error in the pattern p.

Yd^p is the desired output in the pattern p.

Y^p is the actual output in the pattern p.

P is the patterns number.

Z is the parameter vector (a_{11}, …, a_{nm}; b_{11}, …, b_{nm}; c_1, …, c_n).

Fig. (**2**) shows the structure of the identification problem with NF model [2, 4].

The objective function *E(Z)* with respect to all parameters in NFS, such as the centers a_{ij} and the widths b_{ij} for the input membership functions, and the centers c_i for the output membership functions.

$$a_{ij}(t+1) = a_{ij}(t) + k_a\left[\left(Yd(t)-Y(t)\right)\times\left(c_i(t)-Y(t)\right) \times \frac{u_i}{\sum_{L=1}^{n} u_L} \times \frac{\left(x_j - a_{ij}(t)\right)}{\left(b_{ij}(t)\right)^2}\right] \quad (7)$$

$$b_{ij}(t+1) = b_{ij}(t) + k_b\left[\left(Yd(t)-Y(t)\right)\times\left(c_i(t)-Y(t)\right) \times \frac{u_i}{\sum_{L=1}^{n} u_L} \times \frac{\left(x_j - a_{ij}(t)\right)^2}{\left(b_{ij}(t)\right)^3}\right] \quad (8)$$

$$c_i(t+1) = c_i(t) + k_c\left[\left(Yd(t)-Y(t)\right)\times \frac{u_i}{\sum_{L=1}^{n} u_L}\right] \quad (9)$$

Where: i = 1,…, m ; {m is no. of input variables}

j = 1,…, n ; {n is no. of rules in NFS}

k_a, k_b and k_c are the learning rates, and t means the learning iteration.

The determination process of the initial values for the centers a_{ij} of the membership functions is very difficult. The NF network system provides easy way to minimize the difficulty of finding the centers a_{ij}, in this study, the Matlab program is used to solve the above equations as shown in Appendix A.

3. DATA USED IN THE STUDY

This paper is a study of Beijing-Shanghai high-speed railway and measuring the settlement values of the Bridge with the time and load. The settlement is measured using the single point of settlement account meter. The data used in this study were taken from level gauge measurements of pile foundations processed of Beijing-Shanghai project at location DK124, piers D18 and D19 and at DK152, piers F371, F372 and F373 on 2009/10/27. The collected measurements included the displacement/mm and the relative amount of compression/mm. These values were measured for multiple depths and at different time intervals. The reading depths were different depending on the depth of the compressed layer under the pile. The study analysed these data in the NF network system to compare them with the hyperbolic and the statistical models in addition to predict the settlement as shown in the following items.

4. SETTLEMENT OF PILE FOUNDATION

In bridge engineering, settlement of pile-group foundation is an important index reflecting construction safety and quality [9]. Calculation methods of settlement adopted by most of criteria are usually conservative and theory basis is scant and deficient [10]. For pile foundation, however, due to the difficulty in evaluating the interaction of pile-soil-pile system and the behavior of excess pore water pressure, quantitative prediction method for long-term settlement in soft ground soil still needs to be improved [11]. For performance-based design of pile foundations, it is necessary to develop a practical prediction method for long-term displacements of pile foundation. Many prediction methods for the settlement in soft ground can be found, current methods of settlement calculations show different advantages and disadvantages [12-15].

In this paper, the new monitoring technique by a single point of settlement level meter is used to measure the settlement in the deep soft soil. Calculation and prediction of pile group foundation settlement is still a research problem. The NFS models are used to predict the settlement of the

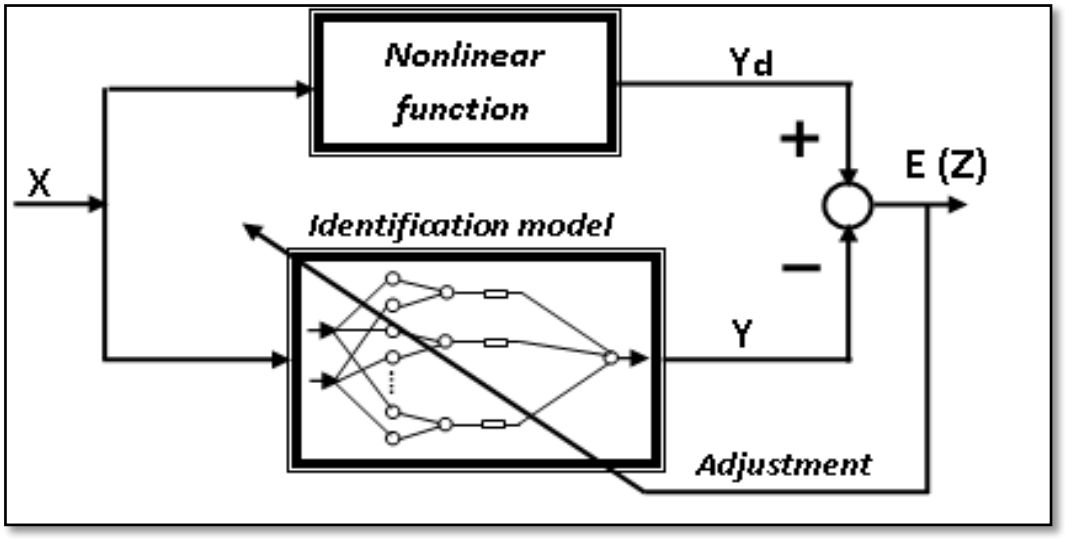


Fig. (2). Identification problem with NeuroFuzzy model.

deep soft soils in the project with high accuracy compared with the previous methods.

Pile foundation is a great achievement of modern theory of interaction between pile and soil. There are two reasons why a pile foundation is more appropriate than a shallow foundation in a certain project. First is because of a low bearing capacity of the natural subsoil, in which piles are used to transfer the load to deeper soil layers or directly to hard bearing stratum. The second is the foundation cannot fulfill the requirements of tolerable deformation, and piles are used to reduce excessive settlement [16].

In this paper, the pile foundation is used with length reached to 52 m to control the settlement in the deep soft soil area. The results show that the maximum settlement values measured during field testing are within the allowable limits by standard specifications. The TB10002.5-2005 code (3.2.1) specifies an allowable settlement of 40-80 mm [17], and the code "200 kilometres per hour passenger railway interim design provisions" specifies an allowable settlement of 50 mm [18]. Ballast railways have two requirements: the allowable settlement of a single foundation must be no more than 80 mm, while two adjacent foundations may have settlement of no more than 40 mm. The AASHTO code also has a limitation for settlement in the pile foundation not exceeds the value of 25 mm [19].

5. SETTLEMENT ANALYSIS WITH NEUROFUZZY MODEL

Neurofuzzy networks are modified recently in the field of geotechnical engineering. The field data are learned by the NF model using the trial and error of the Matlab program coding. The learning rates are discovered by many trials, then the final learning rates which estimated the values of settlement are ka = 0.00955; kb = 0.00907; kc = 0.00091; as shown in Appendix A. These learning rates give a good estimation for settlement, the estimated curve as shown in Fig. (**3**) for D18 of DK124 worksite, which indicates a matching results of that measured in the field. The final results of the model show also that the predicted settlement will be stable and within the final readings of the field measurements. In the same manner, the other models of Beijing-Shanghai project are tested using the NF model with good results as shown in Fig. (**4**) for D19 pier of DK124 worksite, and Figs. (**5**-**7**) for piers F371, F372 and F373 respectively of DK152 worksite. NF model can predict the values of settlement as shown in the following graphs using the programs coding in Appendix A, through applying long period of time out of the range of the learning data.

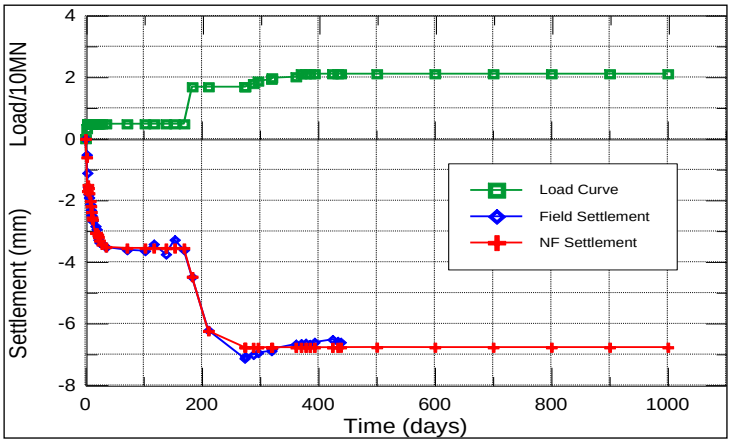


Fig. (3). NF model compared with the field data of D18 pier, DK124.

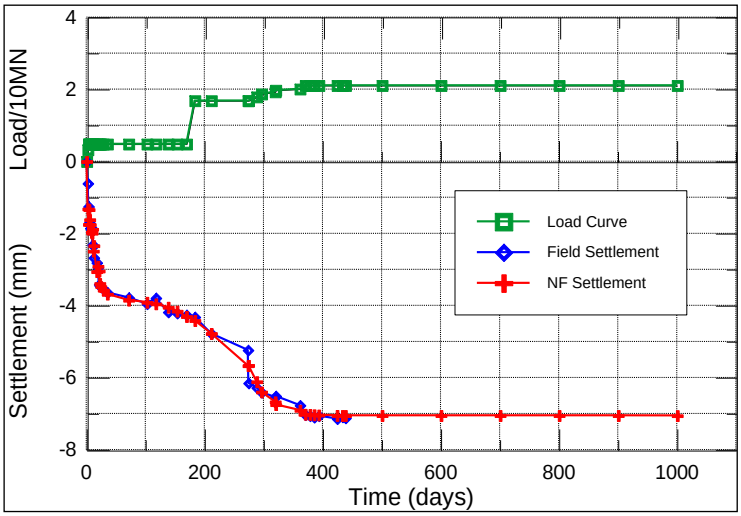


Fig. (4). NF model compared with the field data of D19 pier, DK124.

5.1. Comparison of Neurofuzzy Model with Hyperbolic Model

The settlement can be calculated at any time after loading completion through the use of the following equation [20].

$$S = S_0 + (t / (\alpha + \beta \times t)) \quad (10)$$

Where: S is settlement amount at time t in mm, S_0 is initial settlement amount (at the time of completion of girder construction according to the field measurements) in mm, t is

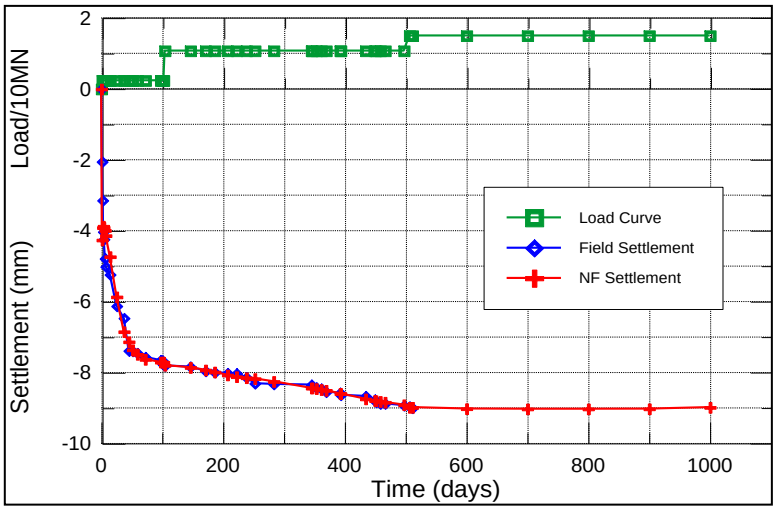


Fig. (5). NF model compared with the field data of F371 pier, DK152.

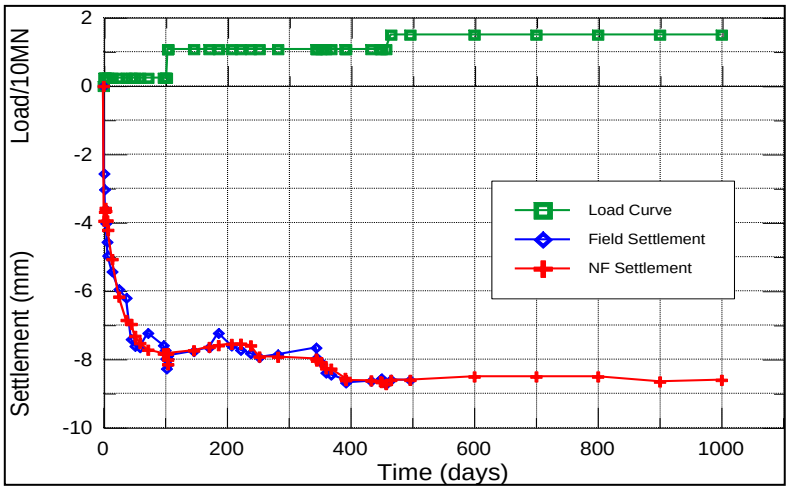


Fig. (6). NF model compared with the field data of F372 pier, DK152.

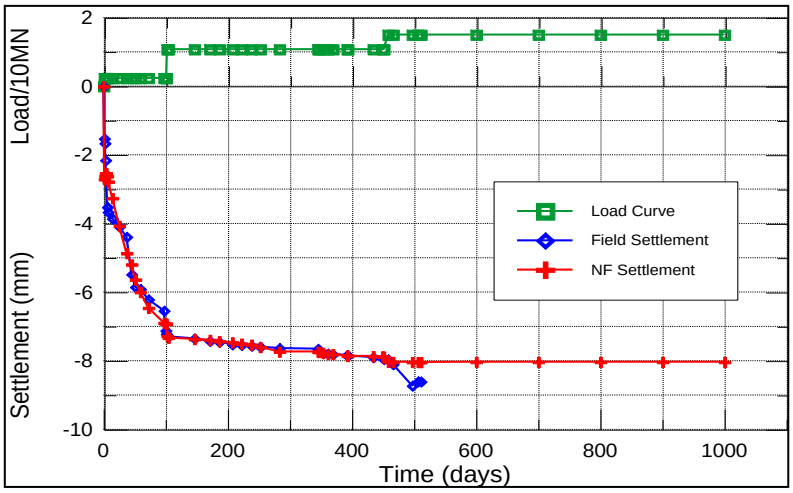


Fig. (7). NF model compared with the field data of F373 pier, DK152.

time in days and α, β are coefficients. After a straight line is drawn, the coefficients α and β are calculated for the vertical axis and slope, respectively, as shown in Fig. (**8**).

Values of α = 8.394 and β = 0.505 are estimated according to the data for pier 18, and pier 19 have values of α = 30 and β = 0.25. The piers F371 and F373 have coefficient values of α = 80, β = 0.761 and α = 45, β = 1.29, respectively. As shown in Figs. (**9**-**12**), the settlement is predicted using the hyperbolic model which assumes an average settlement speed used to predict the settlement based on initial measured settlement amounts. The hyperbolic method is useful for settlement prediction in complex soil formations. However, the precision of the predicted settlement using the hyperbolic model is less than that predicted by the NF model, therefore, the last gives more agreement of the settlement prediction in the deep soft soil. The hyperbolic prediction of settlement for other models of

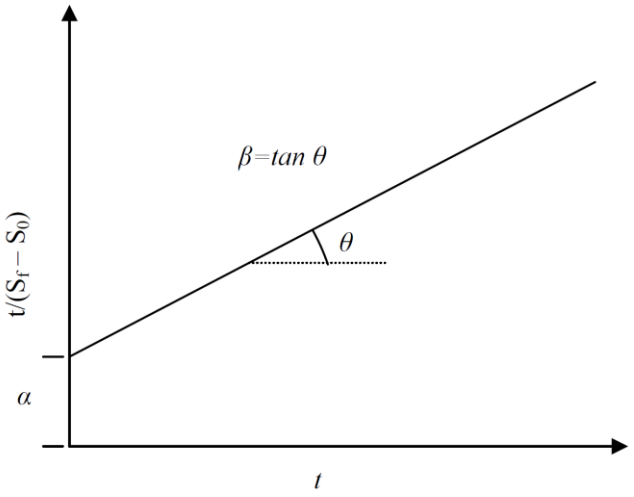


Fig. (8). Hyperbolic function curve coefficients calculation.

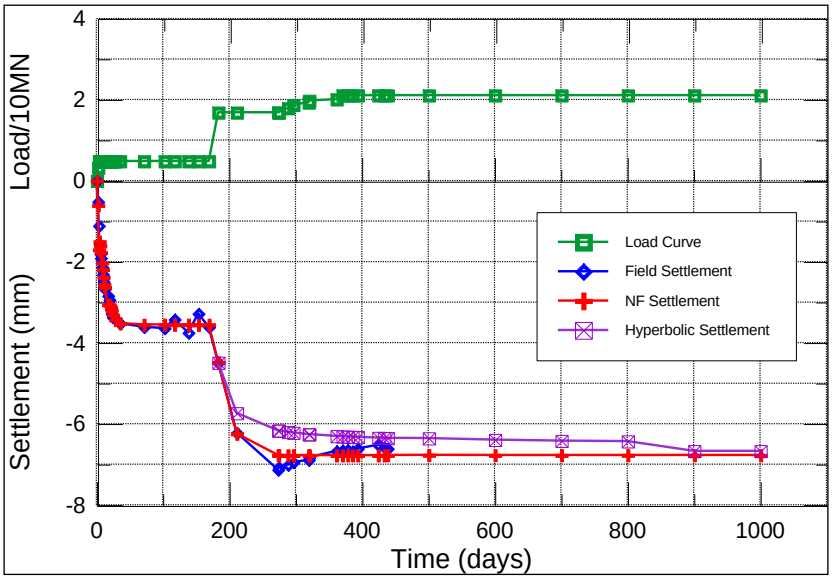


Fig. (9). Hyperbolic model compared with the NF model and the field data of D18, DK124.

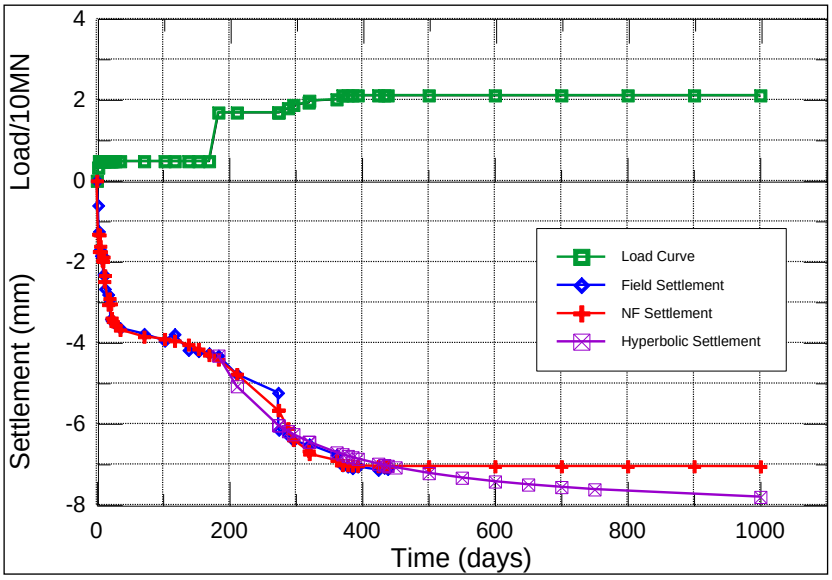


Fig. (10). Hyperbolic model compared with the NF model and the field data of D19, DK124.

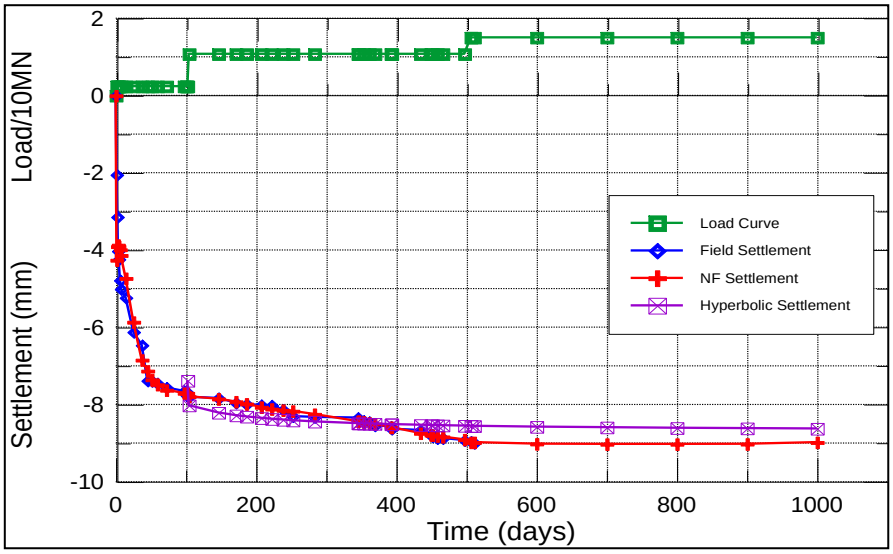


Fig. (11). Hyperbolic model compared with the NF model and the field data of F371, DK152.

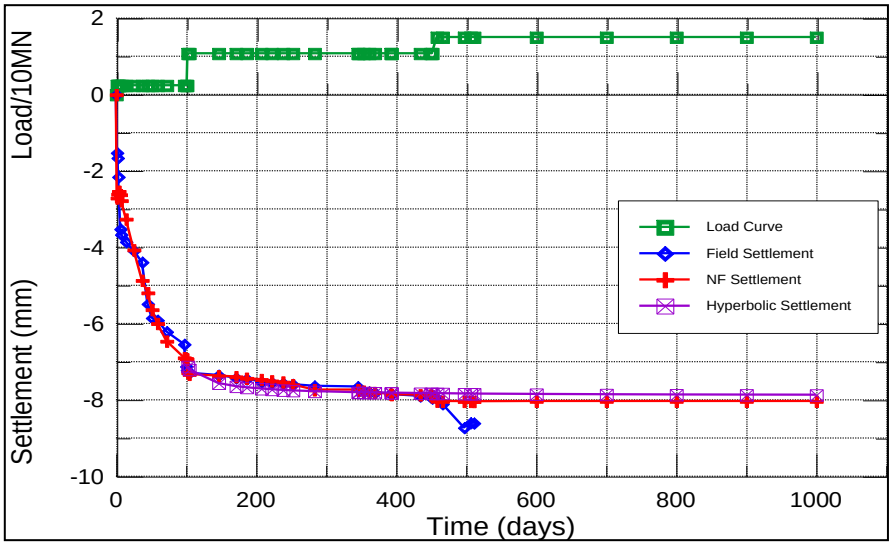


Fig. (12). Hyperbolic model compared with the NF model and the field data of F373, DK152.

Beijing-Shanghai project is not conservative; meanwhile, it is conservative for all the models using the NFS in the analysis and prediction of settlement.

5.2. Comparison of Neurofuzzy Model with Statistical Model

The SAS software is used to derive the settlement prediction equation by considering time and load as independent variables and settlement readings as dependent variable [21]. Appendix B shows the log file of the input program which is used to find the run results. The following equation is derived for pier 18:

$$S_{t18} = -0.812 - 2.674\,L - 0.015\,t + 0.007\,Lt \qquad (11)$$

Where: S_{t18} is settlement value for pier 18 at time t in mm, L is the load/10MN and t is time in days.

The first parameter of the above equation constitutes the initial value of settlement in the statistical analysis. Fig. (**13**) shows the comparison between the predicted values of settlement of NF model and the statistical model of D18 in DK124 worksite, in addition to that predicted by the hyperbolic model. For pier F372 at worksite DK152, the statistical equation is derived from the field data as shown in the following:

$$S_{tF372} = -4.291 - 2.771\,L - 0.011\,t + 0.007\,Lt \qquad (12)$$

Fig. (**14**) shows also the relation between the NFS prediction with the statistical regression of F372 in DK152. The NFS prediction gives good agreement compared with the other models and can be applied for similar problems to estimate the settlement of pile foundations. The statistical regressions of other models of the project are not conservative as the same manner in the hyperbolic model; therefore, the prediction of settlement is conservative using the NFS. More information about the statistical analysis and Eureqa Models are shown in Appendix C and Appendix D respectively. As shown from the results of Eureqa Models,

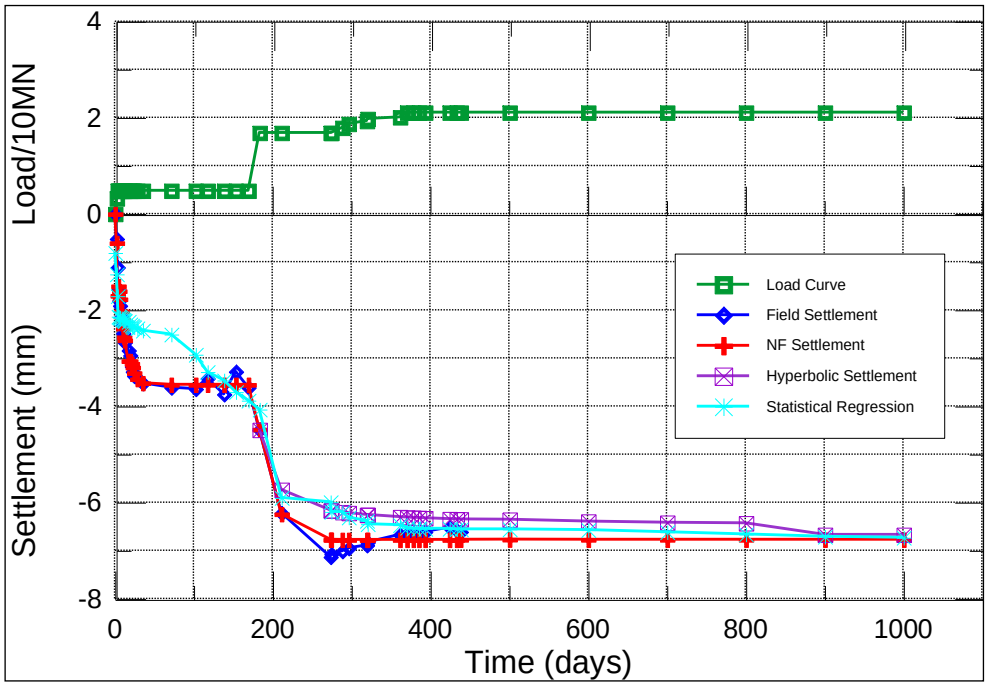


Fig. (13). NF model compared with hyperbolic model, statistical model and the field data of D18, DK124.

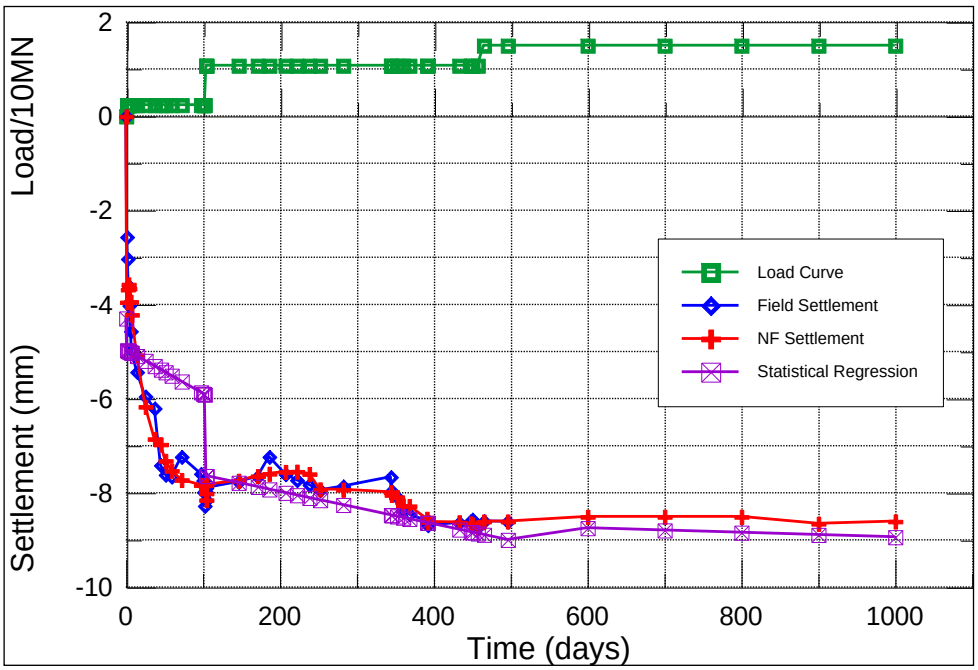


Fig. (14). NF model compared with hyperbolic model, statistical model and the field data of F372, DK152.

the data behaves in a good statistically manner. Observed *vs.* Predicted, Output *vs.* Row and Error/Complexity Pareto are shown in Eureqa Models which shows also the behaviour of data with stable curves and changeable behaviour that indicate the reasonability of the settlement and not its reaching to the dangerous limits. When compared NFS Models with Eureqa Models, the first shows more stable state behaviour and long duration behaviour, NFS is more suitable than other methods to predict the settlement of pile foundations.

CONCLUSION

In recent years, NF networks have emerged as one of the most potentially successful modeling approaches in engineering. In particular, NFS have been applied to many areas of geotechnical engineering and have demonstrated considerable success. The objective of this study is to highlight the use of NFS in pile foundation engineering. The study describes NF techniques and its application in pile foundations settlement, as well as the salient features associated with NF model development. Therefore, the NF

model gives more accurate results to predict the settlement of deep pile foundations compared with other modeling approaches.

Experimental study of the issues raised by a single point of the settlement account of level joint test method measured by a series of compression curve and the total bridge pile foundation settlement. For the application of theoretical and engineering significance of compression settlement point by adjacent to a single point, the NFS gives good estimations of settlement compared with the field data.

CONFLICT OF INTEREST

The author confirms that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

We thank all the project team of Beijing-Shanghai High Speed Railway to their efforts of collecting the data and the great job to finish this project.

Appendix

APPENDIX A: THE NEUROFUZZY MODEL FOR LEARNING THE DATA

1 First program

```
clear all
clc
memb=5;         % these are the membership functions values
iteration=5000;
error = 0.0001;
spreat=0.028;
ka = 0.00955;  kb = 0.00907;    kc =0.00091;

%-------------------------------------------------------------------------
data = xlsread('data.xls');     % read the Xls file and divide it to the 3 variables Time (t), Load (l), and Settlement (s)
t= data(1:length(data), 1);
l= data(1:length(data), 2);
ste= data(1:length(data), 3);

pattren=length (ste);
max_t= max(t);
max_l= max(l);
max_ste= min(ste);     % we take minimum because the values of Settlement is Negative
                % Transfer all the Data to 0..1 depend on
                % its values and in the end of program we
                % will return the values
t=t/max_t;
l=l/max_l;
ste=ste/max_ste;
%-------------------------------------------------------------------------
% these are the high and low values for Time, Load and Settlement.
high_t=max(t); low_t=min(t);
high_l=max(l); low_l=min(l);

high_s=max(ste); low_s=min(ste);
%-------------------------------------------------------------------------
% calculate the increment for all inputs
inctm=(high_t-low_t)/(memb-1);
inclm=(high_l-low_l)/(memb-1);

incsm=(high_s-low_s)/(memb-1);

% calculate the centers for all input variables
centt=low_t:inctm:high_t;
centl=low_l:inclm:high_l;

cents=low_s:incsm:high_s;
```

```
b1(1:memb)= spreat;
b2(1:memb)= spreat;
%-------------------------------------------------------------------
% hold on

e=999999;
it=0;
inter_arr=[];
error_arr=[];
while (e > error) && (it < iteration)
  clc
  e=0;
  for i=1:pattren
    s=0;

    for j=1:memb
      q1= -0.5 * ( (t(i) - centt(j)) / b1(j) ).^2;
      if (q1 <= -32)
        u1(j)= 0;
      else
        u1(j)= exp (q1);
      end;

      q2= -0.5 * ( (l(i) - centl(j)) / b2(j) ).^2;
      if (q2 <= -32)
        u2(j)= 0;
      else
        u2(j)= exp (q2);
      end;

      u_pr(j)= u1(j) * u2(j);
      s= s + u_pr(j);
    end

    for j=1:memb
      if s < 0.000000000001
        u_bar(j)= 0;
      else
        u_bar(j)= u_pr(j) / s;
      end
    end

    oy(i)= 0;
    for j=1:memb
      oy(i)= oy(i) + cents(j) * u_bar(j);
    end

    e= e + 0.5 * (ste (i) - oy(i)).^2;

  %% Back propagation algorithm
    for j=1:memb
      kkk= (oy(i) - ste(i)) * (cents(j) - oy(i)) * u_bar(j);
      ea1= kkk * ( (t(i) - centt(j)) / (b1(j)).^2 );
      ea2= kkk * ( (l(i) - centl(j)) / (b2(j)).^2 );

      eb1= kkk * ( (t(i) - centt(j)).^2 / (b1(j) * (b1(j)).^2 ) );
      eb2= kkk * ( (l(i) - centl(j)).^2 / (b2(j) * (b2(j)).^2 ) );

      ec= (oy(i) - ste(i)) * u_bar(j);
```

```
    centt(j)= centt(j) - ka * ea1;
    centl(j)= centl(j) - ka * ea2;
    b1(j)= b1(j) - kb * eb1;
    b2(j)= b2(j) - kb * eb2;
    cents(j)= cents(j) - kc * ec;
   end;

 end

 it= it + 1;
 disp ('Error in Network  =  ');
 disp (e);
 disp ('------------------------');
 disp ('Iteration         =  ');
 disp (it);
%   pause(0.001);
 error_arr=[error_arr log(e)];
 inter_arr=[inter_arr it];
end

% Transfer all the data to 0..1 depend on its values and in the end of program we will return the values.
t=t*max_t;
l=l*max_l;
ste=ste*max_ste;
oy=oy*max_ste;
data(:,1)=t;
data(:,2)=l;
data(:,3)=ste;
data(:,4)=oy;
XLSWRITE('mydata.xls',data);
plot (inter_arr, error_arr);
% legend ('sin (x)');
% xlabel ('x = 0: 2: pi');
% ylabel ('sin (x)  cos (x)');
% title ('plot sin cos function');
% hold
```

2. Second program

```
clear all
clc
memb=5;         % these are membership functions values
iteration=5000;
error = 0.0001;
spreat=0.028;
ka = 0.00951;  kb = 0.00907;    kc =0.0009;


%-------------------------------------------------------------------------
data = xlsread('data.xls');     % read the Xls file and divide it to the 3 variables Time (t), Load (l), and Settlement (s)
t= data(1:length(data), 1);
l= data(1:length(data), 2);
ste= data(1:length(data), 3);

pattren=length (ste);
max_t= max(t);
max_l= max(l);
max_ste= min(ste);              % we take minimum because the values of
                        Settlement is negative
```

```
                    % Transfer all the Data to 0..1 depend on
                    % its values and in the end of program we
                    % will return the values
t=t/max_t;
l=l/max_l;
ste=ste/max_ste;
%-------------------------------------------------------------------------
% these are the high and low values for Time, Load and Settlement.
high_t=max(t); low_t=min(t);
high_l=max(l); low_l=min(l);

high_s=max(ste); low_s=min(ste);
%-------------------------------------------------------------------------
% calculate the increment for all inputs
inctm=(high_t-low_t)/(memb-1);
inclm=(high_l-low_l)/(memb-1);

incsm=(high_s-low_s)/(memb-1);

% calculate the centers for all input variables
centt=low_t:inctm:high_t;
centl=low_l:inclm:high_l;

cents=low_s:incsm:high_s;

b1(1:memb)= spreat;
b2(1:memb)= spreat;
%-------------------------------------------------------------------------
% hold on

e=999999;
it=0;
inter_arr=[];
error_arr=[];
while (e > error) && (it < iteration)
  clc
  e=0;
  for i=1:pattren
    s=0;

    for j=1:memb
      q1= -0.5 * ( (t(i) - centt(j)) / b1(j) ).^2;
      if (q1 <= -32)
        u1(j)= 0;
      else
        u1(j)= exp (q1);
      end;

      q2= -0.5 * ( (l(i) - centl(j)) / b2(j) ).^2;
      if (q2 <= -32)
        u2(j)= 0;
      else
        u2(j)= exp (q2);
      end;

      u_pr(j)= u1(j) * u2(j);
      s= s + u_pr(j);
    end
```

```
   for j=1:memb
    if s < 0.000000000001
       u_bar(j)= 0;
    else
       u_bar(j)= u_pr(j) / s;
    end
   end

   oy(i)= 0;
   for j=1:memb
     oy(i)= oy(i) + cents(j) * u_bar(j);
   end

   e= e + 0.5 * (ste (i) - oy(i)).^2;

 %% Back propagation algorithm
   for j=1:memb
     kkk= (oy(i) - ste(i)) * (cents(j) - oy(i)) * u_bar(j);
     ea1= kkk * ( (t(i) - centt(j)) / (b1(j)).^2 );
     ea2= kkk * ( (l(i) - centl(j)) / (b2(j)).^2 );

     eb1= kkk * ( (t(i) - centt(j)).^2 / (b1(j) * (b1(j)).^2 ) );
     eb2= kkk * ( (l(i) - centl(j)).^2 / (b2(j) * (b2(j)).^2 ) );

     ec= (oy(i) - ste(i)) * u_bar(j);

     centt(j)= centt(j) - ka * ea1;
     centl(j)= centl(j) - ka * ea2;
     b1(j)= b1(j) - kb * eb1;
     b2(j)= b2(j) - kb * eb2;
     cents(j)= cents(j) - kc * ec;
   end;

  end

  it= it + 1;
  disp ('Error in Network  =  ');
  disp (e);
  disp ('------------------------');
  disp ('Iteration         =  ');
  disp (it);
%   pause(0.001);
  error_arr=[error_arr log(e)];
  inter_arr=[inter_arr it];
end

oper(:,1)= centt;
oper(:,2)= centl;
oper(:,3)= cents;
oper(:,4)= b1;
oper(:,5)= b2;
XLSWRITE('operators.xls',oper);

% Transfer all the Data to 0..1 depend on its values and in the end of program we will return the values.
t=t*max_t;
l=l*max_l;
ste=ste*max_ste;
oy=oy*max_ste;
```

```
data(:,1)=t;
data(:,2)=l;
data(:,3)=ste;
data(:,4)=oy;
XLSWRITE('mydata.xls',data);

plot (inter_arr, error_arr);
% legend ('sin (x)');
% xlabel ('x = 0: 2: pi');
% ylabel ('sin (x)  cos (x)');
% title ('plot sin cos function');

% hold off
```

3. Third program

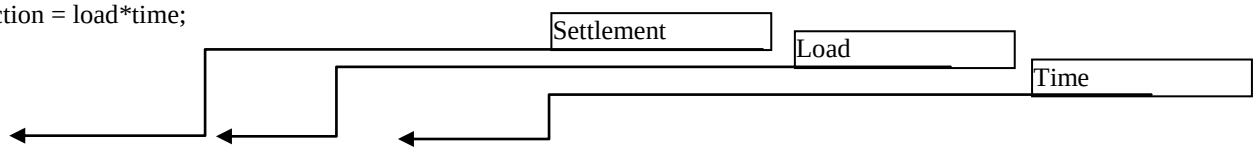
```
clear all
clc
memb=5;          % these are membership functions values
%--------------------------------------------------------------------------
oper= xlsread('operators372.xls');
centt= oper(1:length(oper),1);
centl= oper(1:length(oper),2);
cents= oper(1:length(oper),3);
b1= oper(1:length(oper),4);
b2= oper(1:length(oper),5);

data = xlsread('dataF372.xls');     % read the Xls file and divide it to the 3 variables Time (t), Load (l), and Settlement (s)
t= data(1:length(data), 1);
l= data(1:length(data), 2);
ste= data(1:length(data), 3);

pattren=length (t);
max_t= max(t);
max_l= max(l);
max_ste= min(ste);              % we take minimum because the values of Settlement is Negative
t=t/max_t;
l=l/max_l;
ste=ste/max_ste;
for i=1:pattren
   s=0;
   for j=1:memb
     q1= -0.5 * ( (t(i) - centt(j)) / b1(j) ).^2;
     if (q1 <= -32)
       u1(j)= 0;
     else
       u1(j)= exp (q1);
     end;

     q2= -0.5 * ( (l(i) - centl(j)) / b2(j) ).^2;
     if (q2 <= -32)
       u2(j)= 0;
     else
       u2(j)= exp (q2);
     end;

     u_pr(j)= u1(j) * u2(j);
     s= s + u_pr(j);
   end
```

```
    for j=1:memb
      u_bar(j)= u_pr(j) / s;

    end

    oy(i)= 0;
    for j=1:memb
      oy(i)= oy(i) + cents(j) * u_bar(j);
    end
end

% Transfer all the Data to 0..1 depend on its values and in the end of program we will return the values
t=t*max_t;
l=l*max_l;
ste=ste*max_ste;
oy=oy*max_ste;

data(:,1)=t;
data(:,2)=l;
data(:,3)=oy;
XLSWRITE('18.xls',data);
```

APPENDIX B

Code file of settlement data for pier D18

```
data bridge;
input Settlement load time;
Interaction = load*time;
cards;
```



0	0	0
0	0.1603333	1
-0.52	0.325	2
-1.11	0.481	3
-1.57	0.481	4
-1.8	0.481	5
-1.91	0.481	6
-2.13	0.481	8
-2.31	0.481	9
-2.48	0.481	10
-2.53	0.481	11
-2.67	0.481	12
-2.84	0.481	17
-2.94	0.481	19
-3.17	0.481	21
-3.28	0.481	22
-3.28	0.481	24
-3.37	0.481	28
-3.41	0.481	35
-3.42	0.481	71
-3.52	0.481	102
-3.61	0.481	117
-3.61	0.481	138
-3.64	0.481	153
-3.75	0.481	169
-4.49	1.684	183
-6.22	1.684	211
-6.52	1.684	273
-6.58	1.684	274
-6.61	1.7905	288
-6.61	1.870375	296

-6.64	1.9302813	319
-6.65	1.9752109	320
-6.67	2.0089082	361
-6.68	2.11	370
-6.83	2.11	378
-6.89	2.11	385
-6.95	2.11	393
-7.01	2.11	424
-7.07	2.11	433
-7.13	2.11	438

```
ODS GRAPHICS ON;
PROC univariate Data = bridge;
qqplot time;
run;
PROC Reg Data = bridge;
model Settlement = load time Interaction;
plot predicted.*residual.;
run;
PROC Reg  Data = bridge;
model Settlement = time;
run;
```

APPENDIX C

Data Used in the Study

The data used in this study were taken from level gauge measurements of pile foundations processed at locations DK124 and DK152 that started on 2009-10-27. The measurements collected included the displacement/mm and the relative amount of compression/mm. These values were measured for multiple depths and at different time intervals. The reading depths were different depending on the depth of the compressed layer under the pile. The study represented these data in a graphical form of the experimental results to predict settlement of deep soft soils. Tables C-1 and C-2 show the geotechnical properties and parameters of the soft soil in the DK124 and DK152 worksites respectively.

Table C-1. Soil characteristics and strength parameters at the DK124 working points.

Sampling Depth (m)	ω %	γ_{unsat} (kN/m^3)	γ_{sat} (kN/m^3)	e	k (m/day)	ν	a_v (MPa^{-1})	c (kPa)	φ (Degree)	Ψ (Degree)
1.07	30.7	17.7	18.0	0.849	0.086	0.32	0.51	14	8.5	0.0
4.37	27.4	18.3	18.6	0.813	0.086	0.31	0.17	15.9	9.2	0.0
9.37	31.1	18.2	18.5	0.852	0.432	0.32	0.47	11.0	10.6	0.0
19.57	26.7	19.4	19.5	0.756	0.432	0.30	0.41	25.6	13.5	0.0
24.82	22.9	19.3	19.5	0.651	0.432	0.30	0.25	20.9	16.7	0.0
26.37	24.2	20.4	20.6	0.658	0.864	0.29	0.32	21.4	30.8	0.8
30.72	26.0	18.7	18.9	0.737	0.086	0.29	0.23	96.8	14.6	0.0
41.97	28.2	18.9	19.2	0.804	0.432	0.28	0.31	35.6	15.5	0.0
46.27	24.7	19.2	19.4	0.677	0.432	0.28	0.13	42.0	21.3	0.0
53.07	30.6	19.4	19.7	0.887	0.043	0.27	0.20	43.8	17.1	0.0
57.57	27.4	20.0	20.2	0.772	8.64	0.28	0.29	8.9	32.7	2.7
69.57	18.9	19.4	19.7	0.587	0.043	0.27	0.18	43.8	17.1	0.0
73.17	28.2	20	20.2	0.820	8.64	0.28	0.34	8.9	32.7	2.7
80.67	23.9	19.2	19.5	0.685	0.043	0.27	0.23	44.7	15.5	0.0
83.57	23.2	20.2	20.4	0.673	8.64	0.28	0.24	13.7	36.2	6.2

Where: ω: Water content; γ_{unsat}: The unsaturated unit weight of soil; γ_{sat}: The saturated unit weight of soil; e: Void ratio; k: Permeability; ν: Poisson's ratio; a_v: Compressibility; E_s: Compression modulus; c: Cohesion; φ: Internal friction angle; ψ: Dilatancy angle.

Table C-2. Soil characteristics and strength parameters at the DK152 working points.

Sampling Depth (m)	ω %	γ_{unsat} (kN/m^3)	γ_{sat} (kN/m^3)	e	k (m/day)	ν	a_v (MPa^{-1})	c (kPa)	φ (Degree)	Ψ (Degree)
3.3	30.7	17.7	18.9	0.849	0.086	0.30	0.51	52.5	14.8	0.0
13.7	27.4	18.6	18.9	0.813	0.432	0.30	0.17	16.6	11.2	0.0
16.2	31.1	19.4	19.6	0.852	0.432	0.30	0.47	17.0	13.2	0.0
21.4	26.7	19.7	20.0	0.756	8.64	0.28	0.41	6.8	37.7	7.7
25.5	22.9	19.4	19.6	0.651	0.432	0.30	0.25	17.0	13.2	0.0
28.1	24.2	19.9	20.6	0.658	0.864	0.29	0.32	14.8	29.1	0.0
30.3	26.0	19.1	19.3	0.737	0.432	0.28	0.23	35.8	20.1	0.0
34.2	28.2	18.9	19.2	0.804	0.864	0.29	0.31	31.4	24.5	0.0
38.7	24.7	19.8	20.0	0.677	8.640	0.28	0.13	6.8	37.7	7.7
41.7	30.6	18.3	18.4	0.887	0.086	0.31	0.20	30.0	11.2	0.0
43.4	27.4	19.3	20.2	0.772	0.086	0.29	0.29	18.0	11.5	0.0
51.3	18.9	19.5	19.6	0.587	0.043	0.27	0.18	29.0	18.0	0.0
55.1	28.2	19.0	20.2	0.820	0.086	0.29	0.34	38.5	10.9	0.0
64.7	23.9	19.2	19.4	0.685	0.086	0.29	0.23	42.3	21.8	0.0
73.0	23.2	19.9	20.4	0.673	8.64	0.28	0.24	11.6	39.0	9.0

Settlement Prediction by Hyperbolic Modeling

The settlement can be calculated at any time after loading completion through the use of the following equation (C-1).

$$S = S_0 + (t / (\alpha + \beta \times t)) \quad \text{(C-1)}$$

Where: S is the settlement amount at time t in mm, S_o is the initial settlement amount (at the time of completion of girder construction according to the field measurements) in mm, t is the time in days as explained previously in the paper.

Figs. (**C-1** and **C-2**) clearly show the settlement field measurements of the loads imposed, with the required construction time, in addition to the predicted settlement for a period longer than that of the field test. The results show that long-term settlement will not be large as compared to the site measurements. The predicted settlement values are about 6.5 mm for pier No. 18, and 8 mm for pier No. 19 in the DK124 worksite. The predicted curves closely match the experimental curves and indicate that the structure will not be affected by the long-term consolidation. The initial settlement was considered for this model at the time of beam girder construction. After this initial value, the settlement increased abruptly to reach its maximum predicted value.

As shown in Figs. (**C-3** and **C-4**), the long-term settlement predicted by the hyperbolic model for piers F371 and F373, respectively closely matched the field test results. The piers F371 and F373 have coefficient values of $\alpha = 80$, $\beta = 0.761$ and $\alpha = 45$, $\beta = 1.29$, respectively.

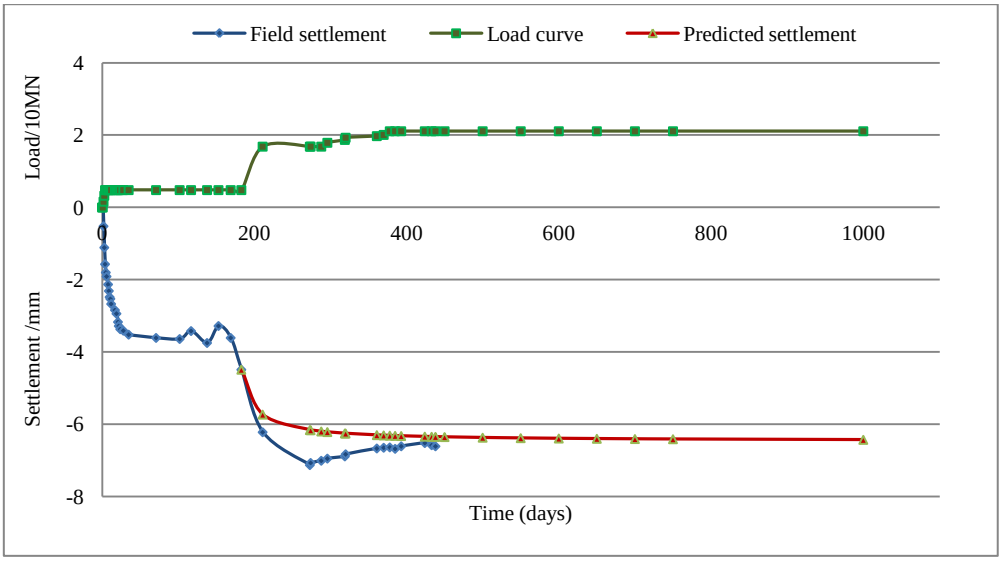


Fig. (C-1). Field settlement, load overtime and predicted settlement curves for pier No. 18.

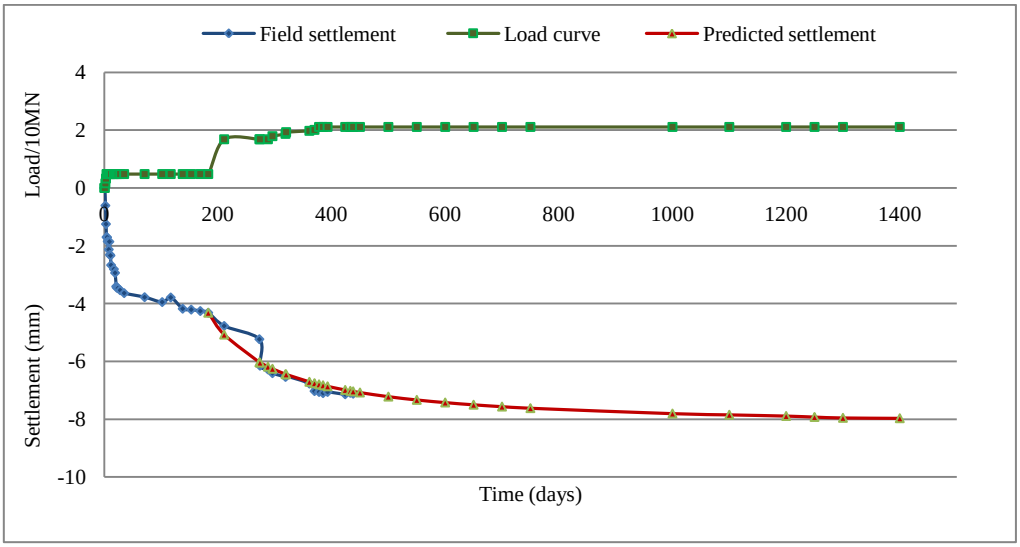


Fig. (C-2). Field settlement, load overtime and predicted settlement curves for pier No. 19.

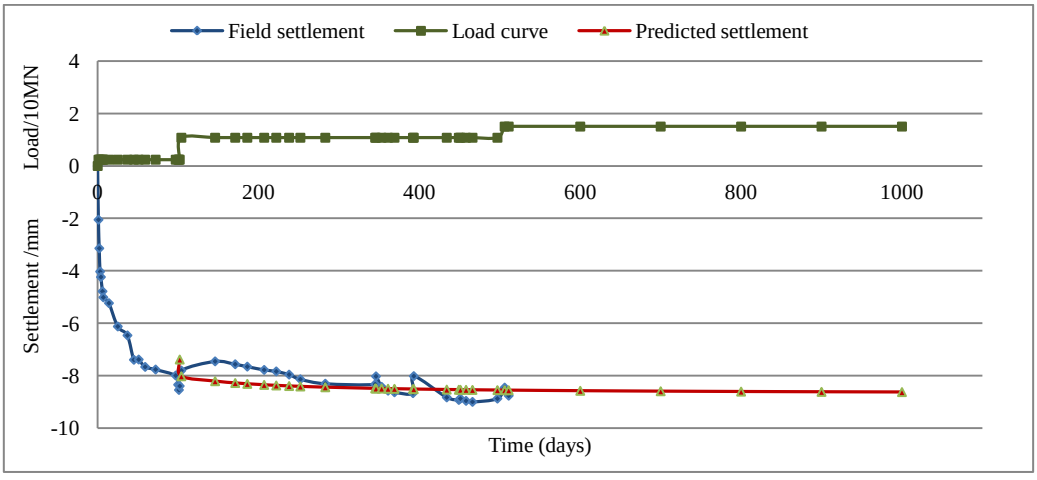


Fig. (C-3). Field settlement, load overtime and predicted settlement curves for pier No. F371.

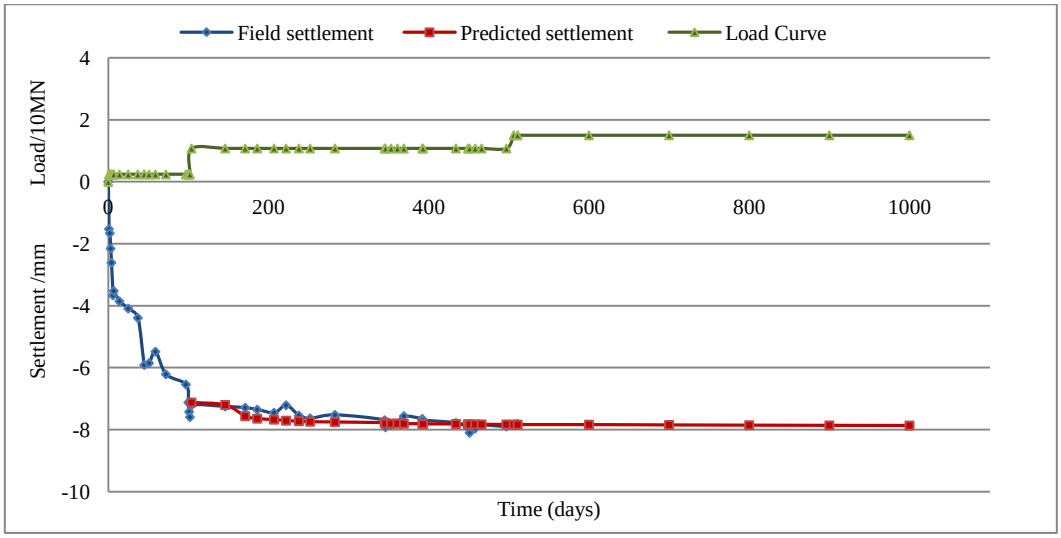


Fig. (C-4). Field settlement, load overtime and predicted settlement curves for pier No. F373.

Settlement Prediction by Statistical Regression

Appendix B shows the log file of the input program that is used to find the output results.

The settlement during the testing time and settlement prediction can be calculated by the equation for pier 18 which mentioned previously in 5-2 as shown in Fig. (**C-5**).

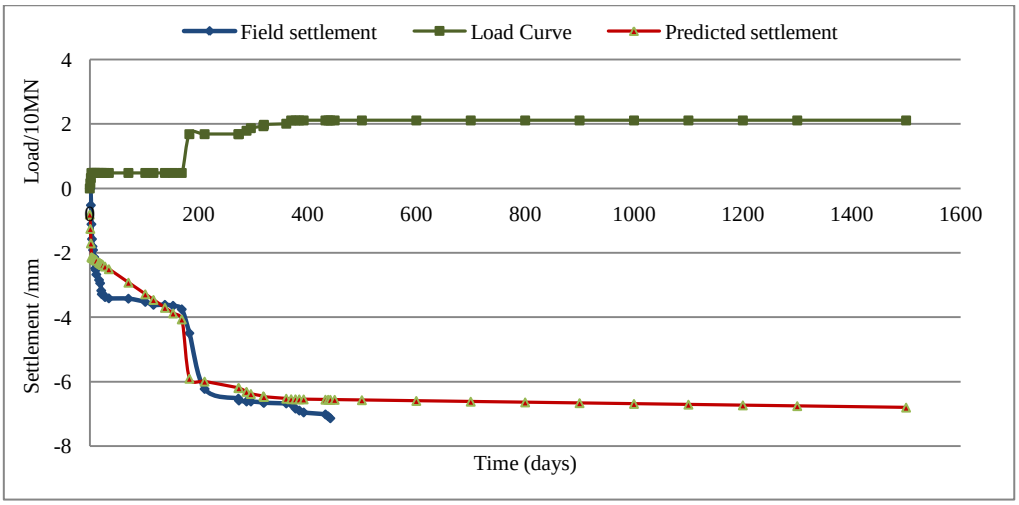


Fig. (C-5). Settlement at pier No. 18 calculated by the statistical regression method.

For pier F372 at worksite DK152 the statistical analysis as shown in Fig. (**C-6**). The predicted settlement has a slight downward trend but remains controlled and within the limits allowed for the long-term settlement. In the statistical analysis, the data have random readings; therefore, they are adjusted for this aspect to get better results and this may be considered as disadvantage in the statistical regression.

From the results obtained in the output of the program shown in Appendix B, the pattern in Fig. (**C-7**) indicates that it may assume that the residuals are not normally distributed and constant in variance at each level of the predicted values. Also, according to QQ-plot analysis, the data are supposed to be normally distributed if the plot forms a straight line, whereas if the plot creates some upward or downward curvatures, the data are supposed to be right-skewness and left-skewness, respectively. In the QQ plot shown in Fig. (**C-8**), the up and down wave form supports indicated that the predicted settlement is not normally distributed. Therefore, the previous discussion explains that the data of the project used in this study are difficult to be analyzed by the statistical models, so the next sections will deal with the three dimensional finite element analysis with new modification to get better results of settlement in the deep soft soils.

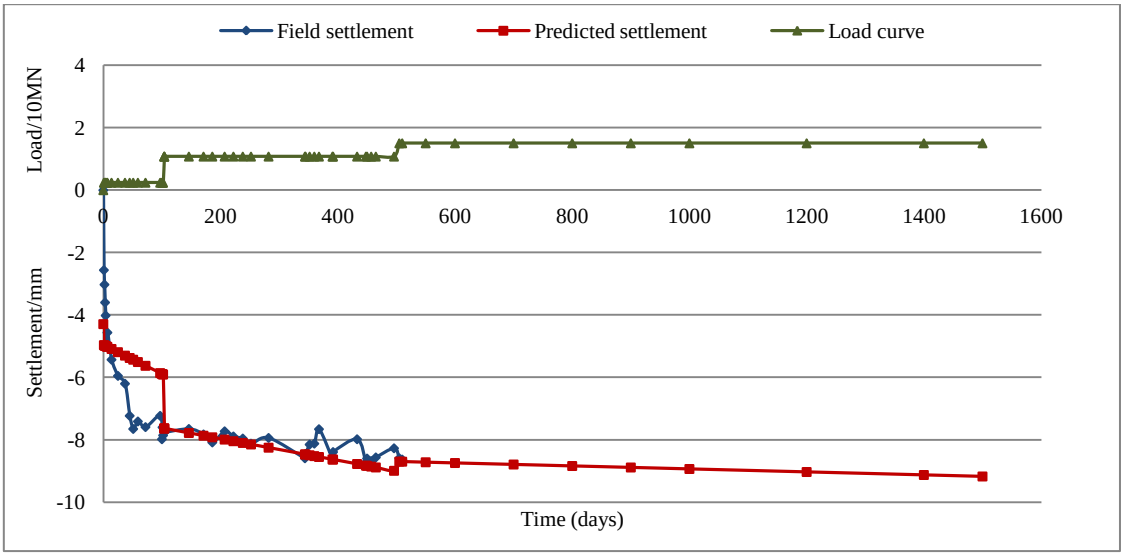


Fig. (C-6). Settlement at pier No. F372 calculated by the statistical regression method.

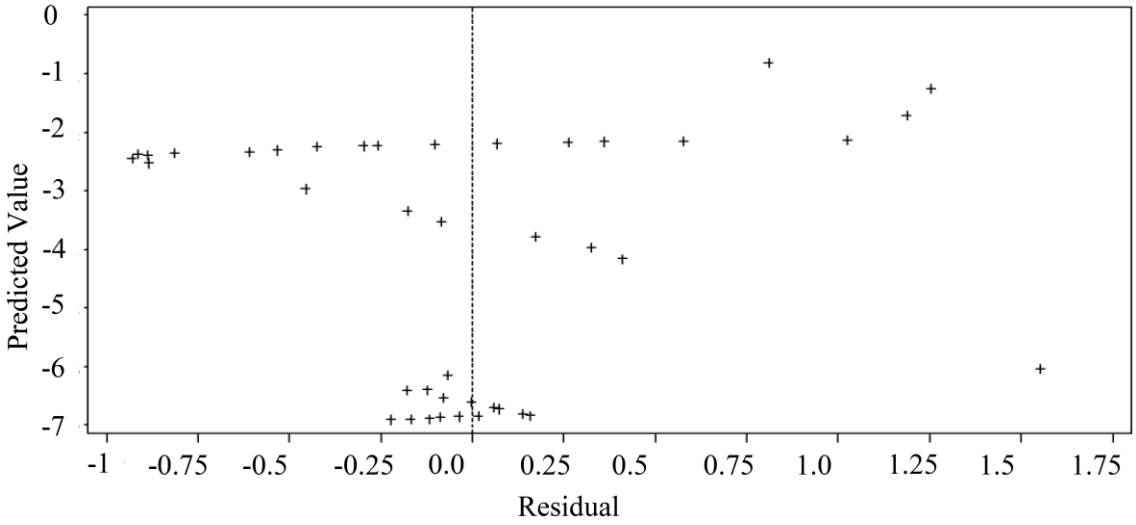


Fig. (C-7). Residual versus predicted value at pier No.18.

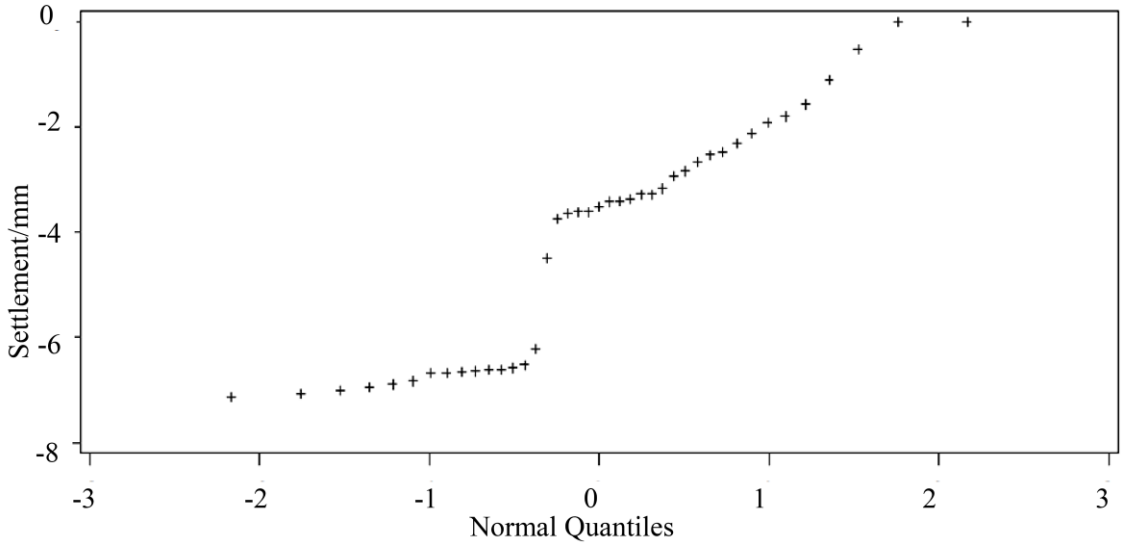


Fig. (C-8). Settlement versus normal quantiles at pier No.18.

The correlation coefficient (R^2) gives more information about the training of network. According to this study the correlation coefficient (R^2) value up to 0.991 for training and testing data sets, and Y = 0.007X+0.041. The results show the ability of an adaptive neuro-fuzzy inference system to predict the damage severity of the structure with high accuracy. Mean Square Error (MSE) is equal to: MSE [T(Y); U] = E [(T(Y) – U)^2], it is equal to 0.003958 as shown in Fig. (**C-9**).

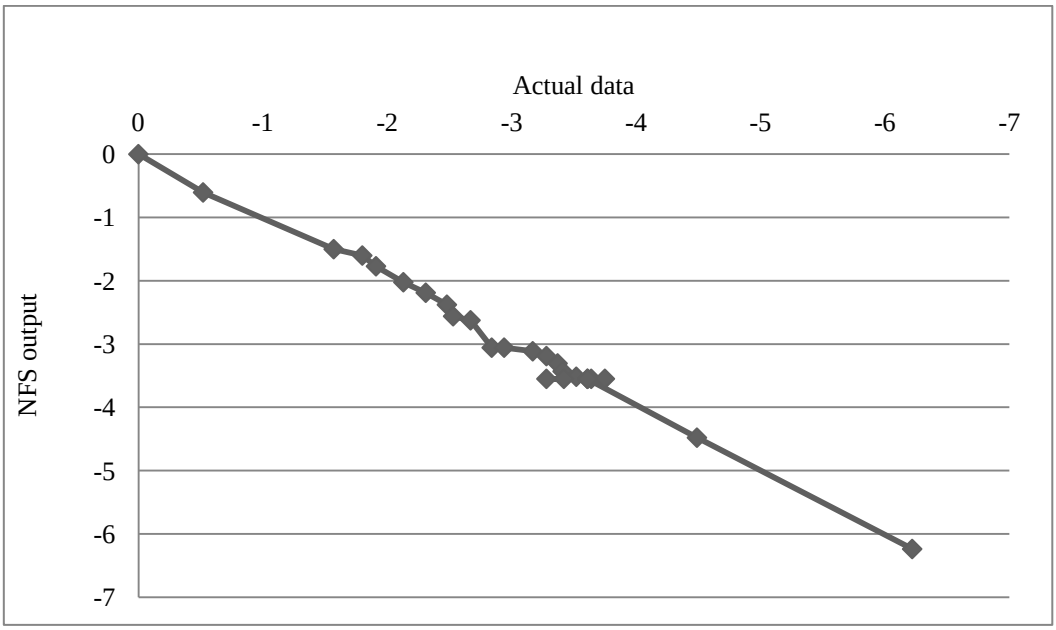


Fig. (C-9). MSE for the training and testing set at pier No.18.

Table **C-3** shows the statistical information of the predicted data by using NFS.

Table C-3. Statistical information of the predicted data by using NFS.

Standard Deviation	Min.	Max.	Mean
2.228711	0	7.13	4.77416

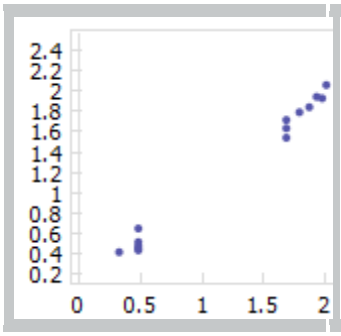
APPENDIX D

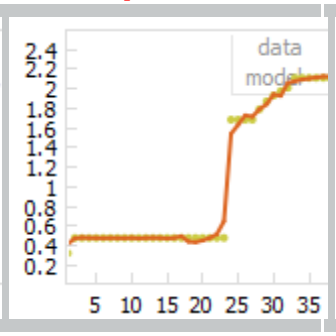
Eureqa Models

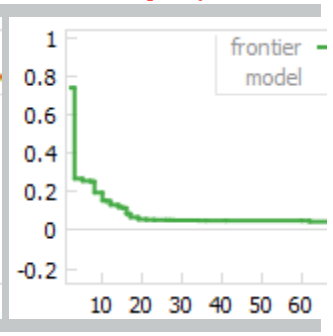
Model List

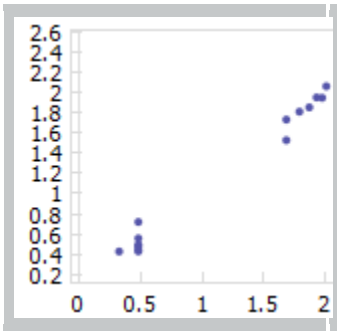
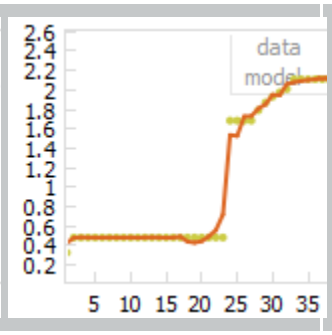
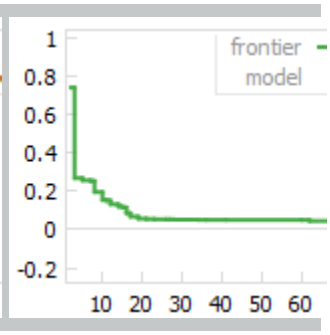
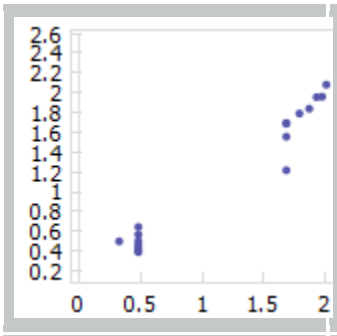
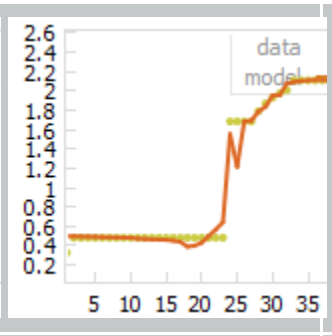
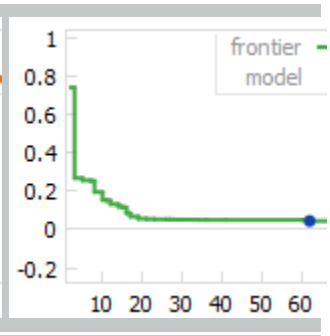
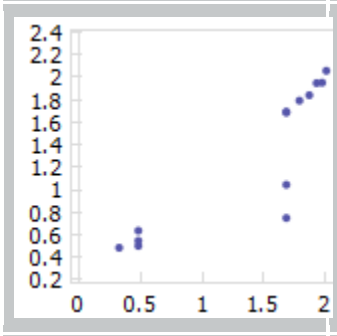
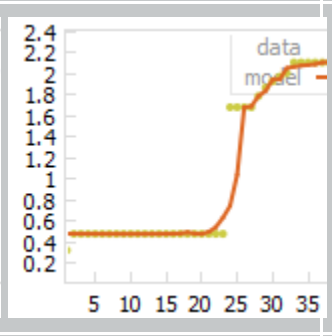
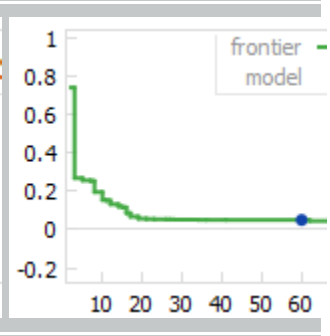
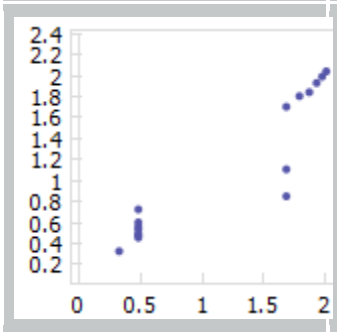
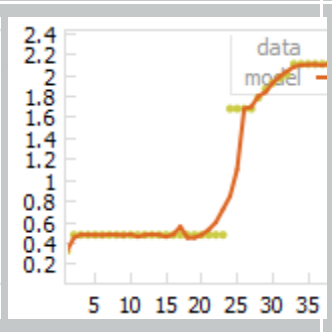
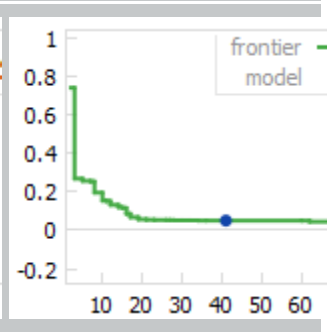
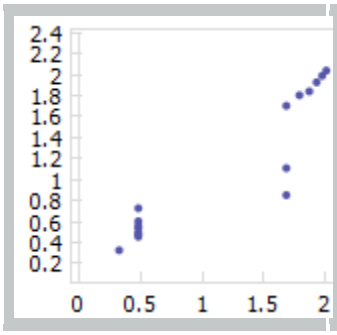
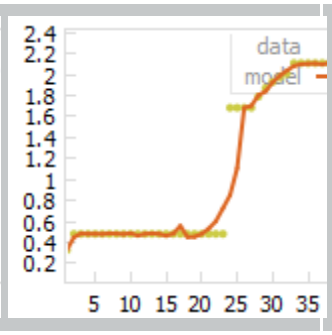
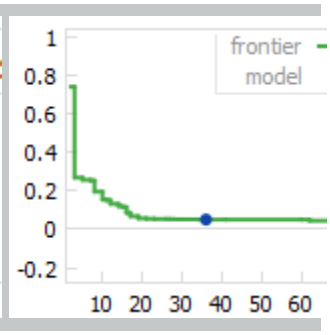
Model	Observed *vs.* Predicted	Output *vs.* Row	Error/Complexity Pareto

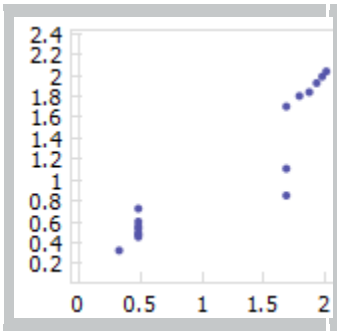
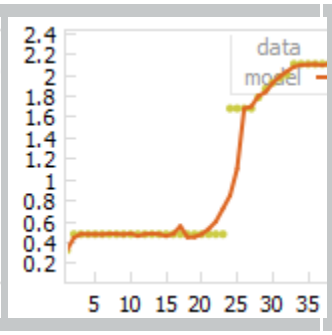
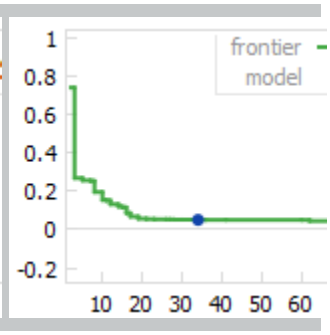
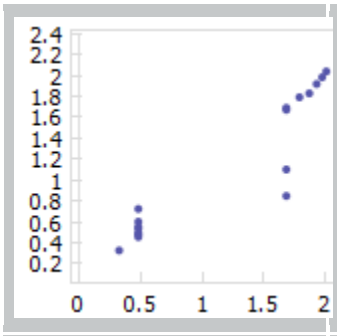
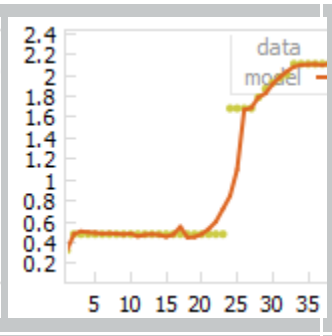
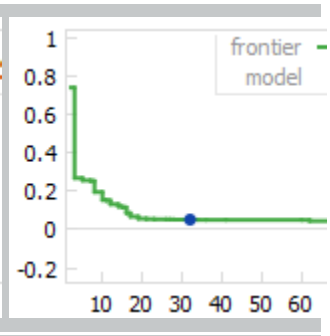
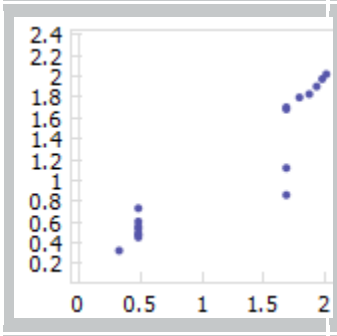
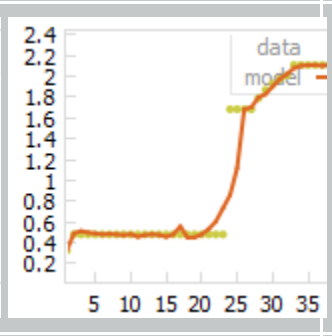
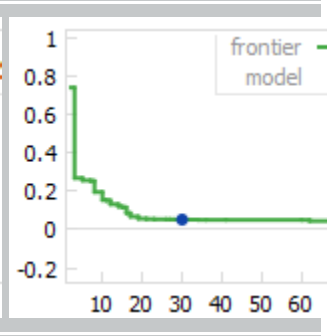
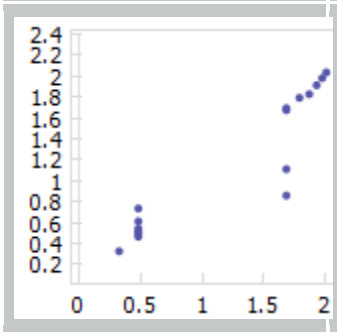
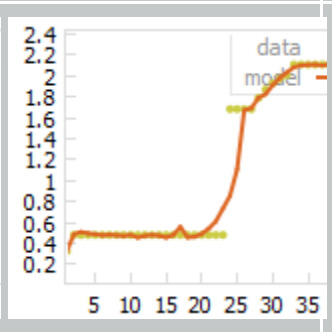
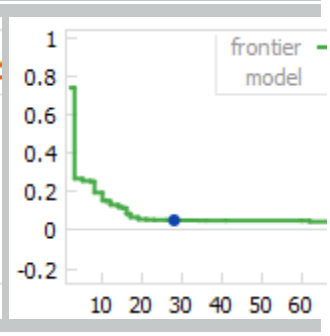
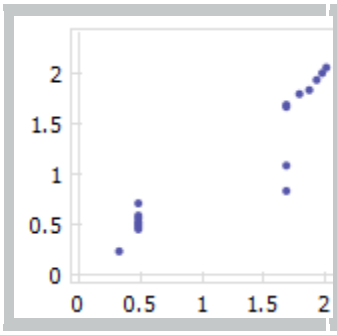
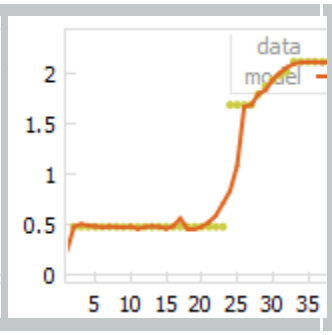
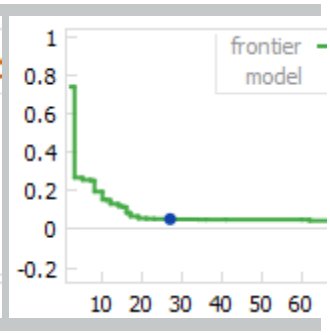
Model:	y = a*cos(b - c*x - d*cos(e + - f/(g*x - cos(-h*w)) - i*x - j*w))
Text:	y = 2.30065*cos(1.45106 - 0.00209287*x - 0.231973*cos(4.88499 + - 0.0416226/(0.00162701*x - cos(-0.351446*w)) - 0.0151917*x - 0.126046*w))

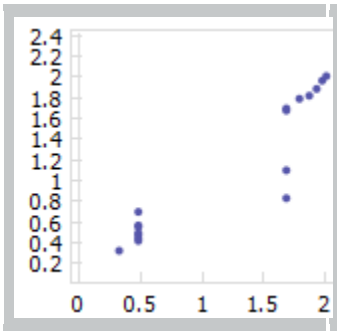
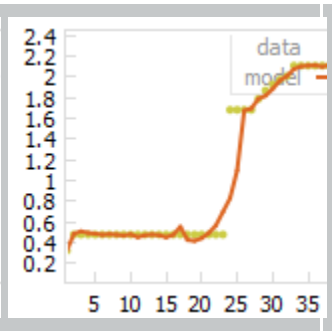
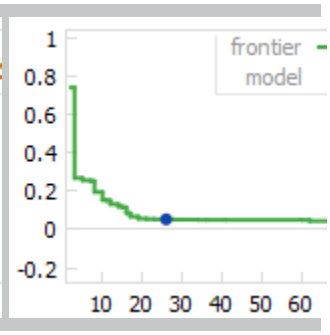
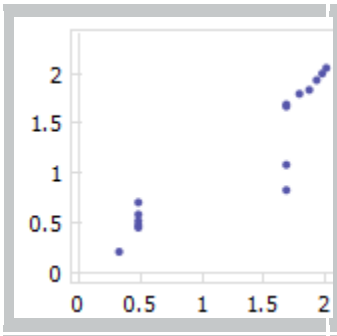
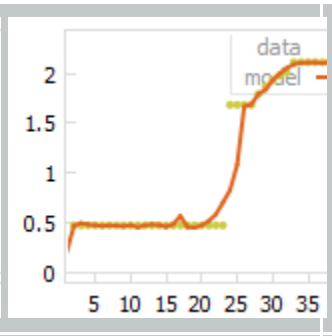
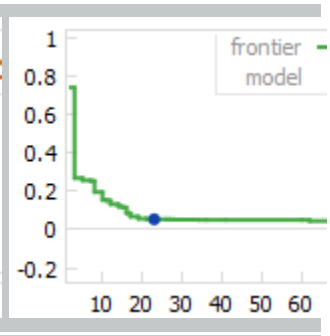
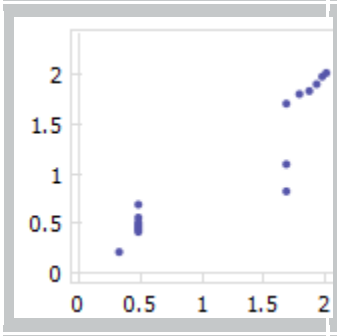
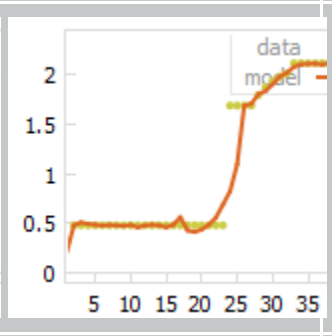
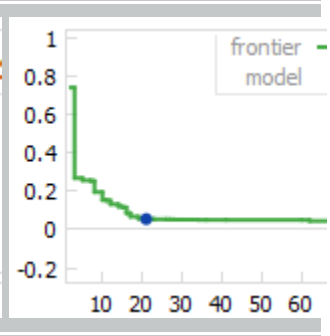
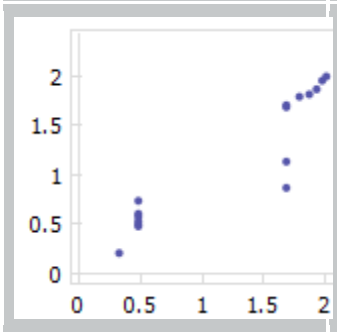
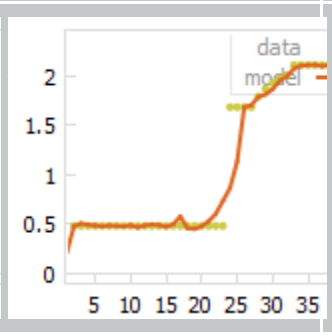
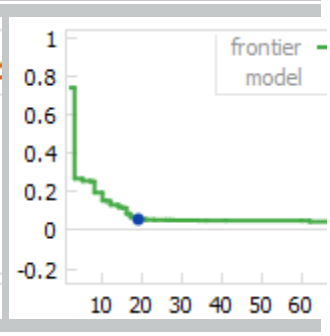
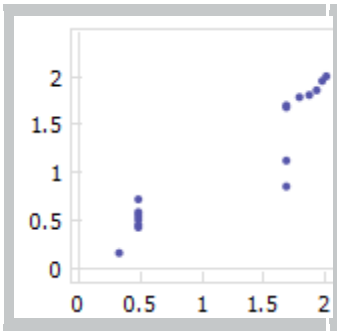
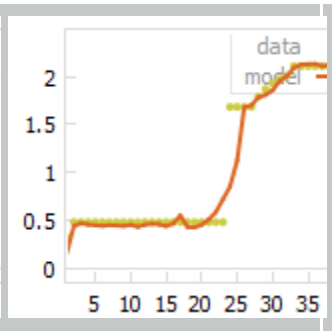
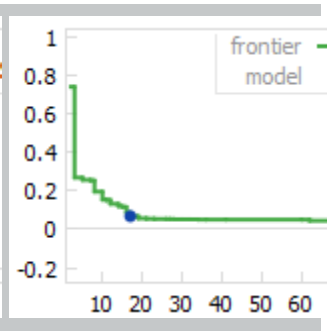


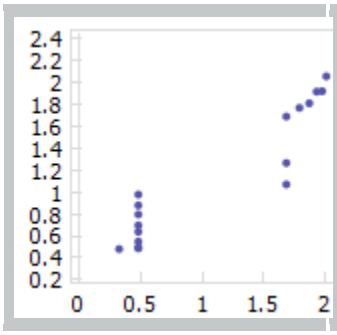
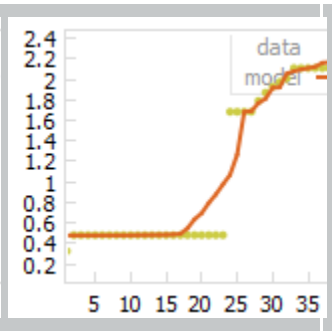
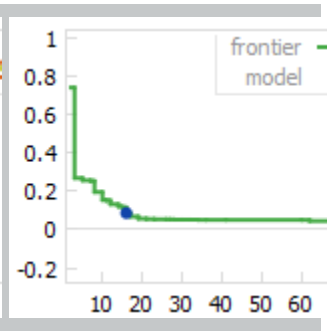
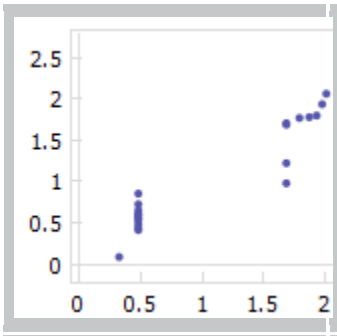
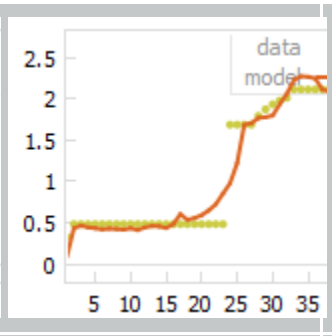
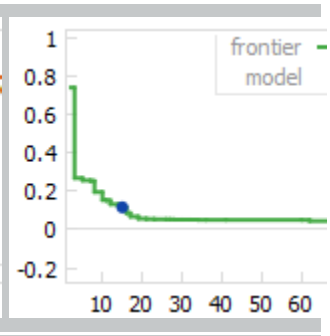
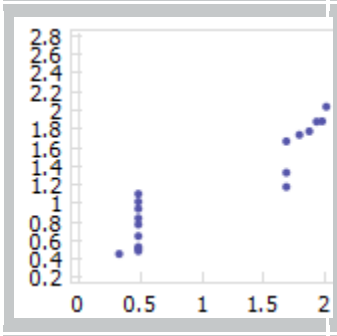
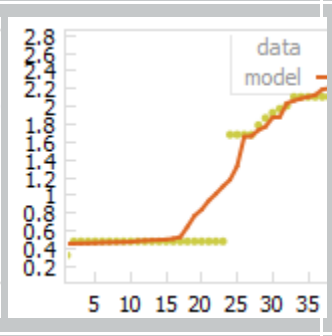
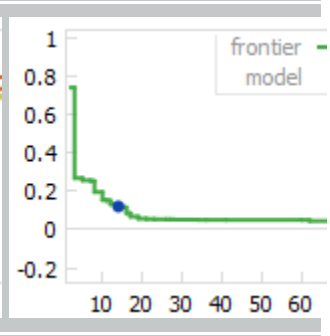
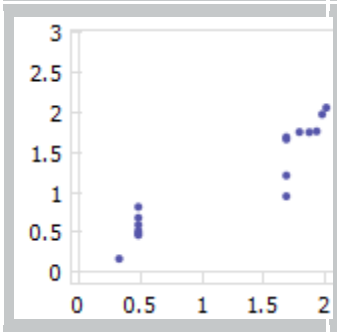
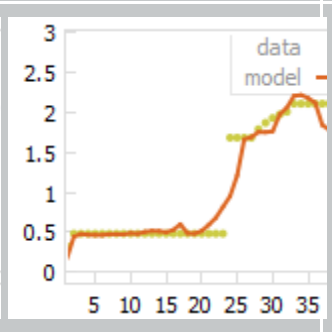
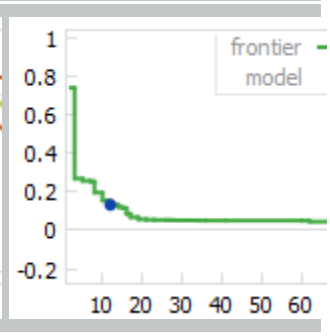
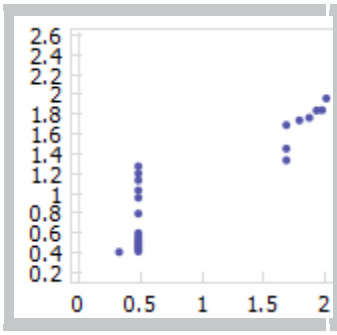
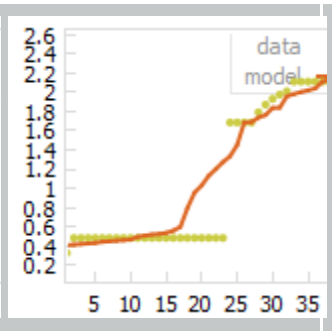
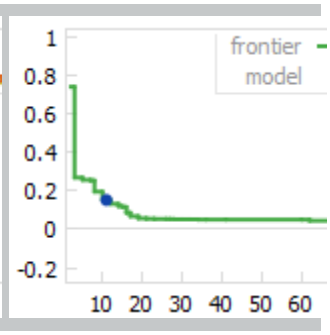


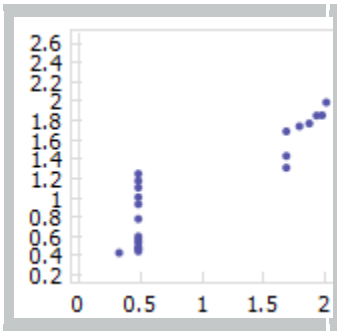
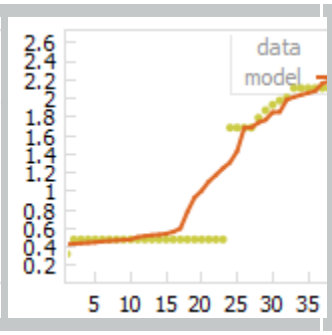
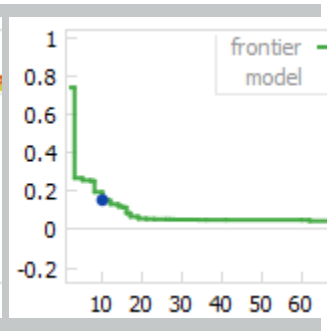
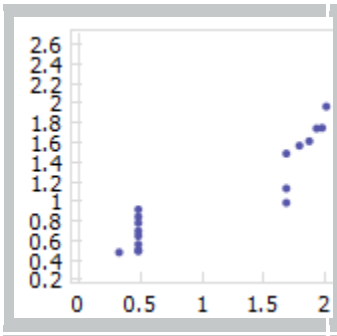
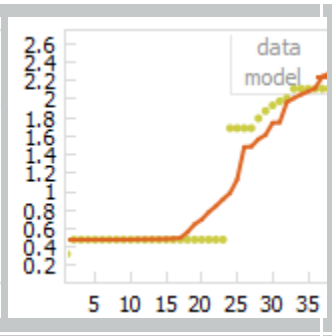
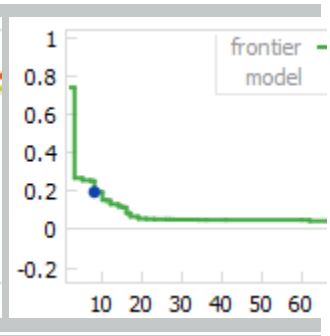
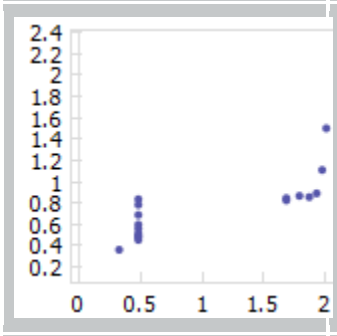
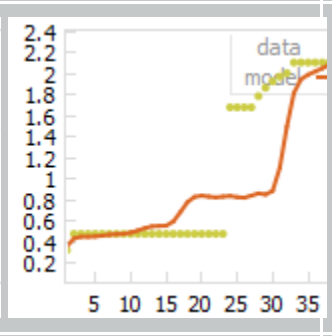
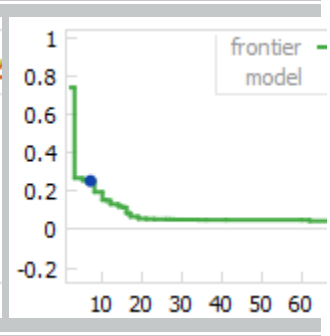
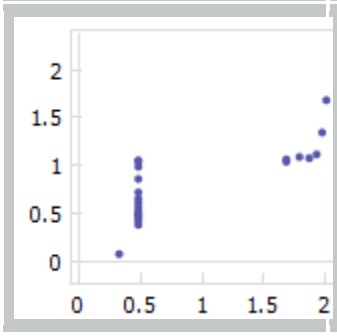
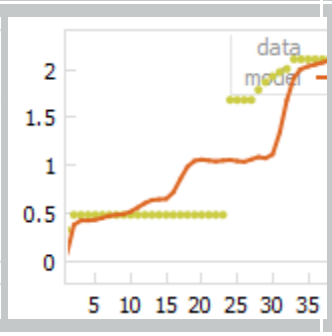
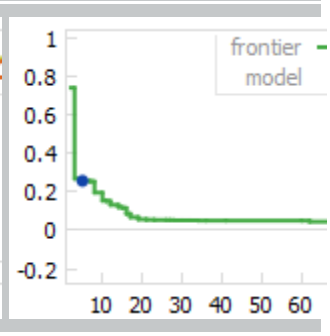
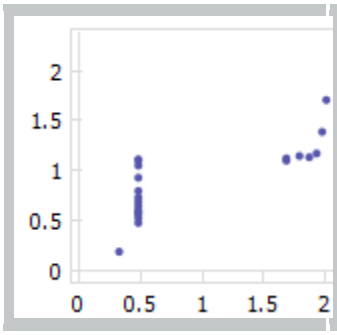
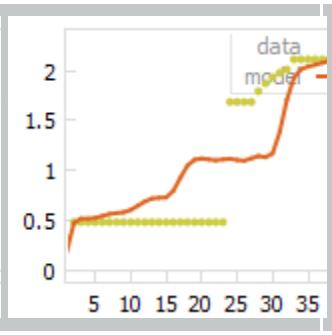
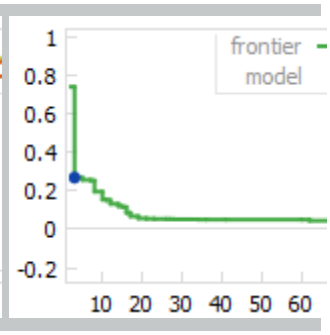


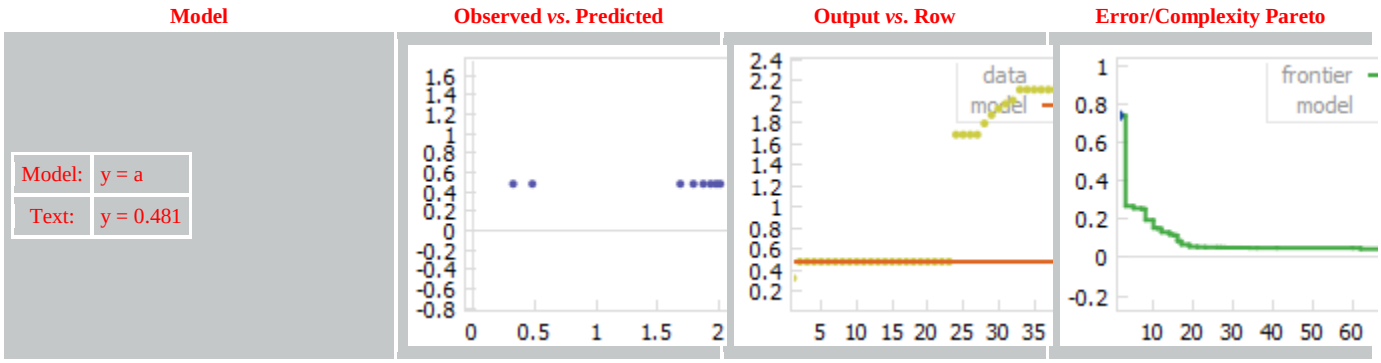
Model	Observed *vs.* Predicted	Output *vs.* Row	Error/Complexity Pareto
Model: y = a*cos(b - c*x - d*cos(e + - f/(g*x - h) - i*x - j*w)) Text: y = 2.31714*cos(1.44466 - 0.00206066*x - 0.225726*cos(4.77843 + - 0.133682/(0.00506785*x - 0.964676) - 0.014796*x - 0.117047*w))			
Model: y = a*cos(b - c*x - d*cos(e + - f/(g*x - h) - i*x)) Text: y = 2.3359*cos(1.43435 - 0.00208401*x - 0.210236*cos(5.05226 + - 0.0468981/(0.00531117*x - 0.984331) - 0.014207*x))			
Model: y = a - b*cos((c + x2 - d*x)/(e*x - f - g*x2)) Text: y = 1.29901 - 0.819268*cos((1689.94 + x^2 - 130.848*x)/(82.5196*x - 17539.8 - 0.330703*x^2))			
Model: y = -a*x/(b*w + x*w - cos(-c*w)) - d*w - e*cos(f*w)*sin(g + h*x) Text: y = -0.384567*x/(2.5413*w + x*w - cos(-2.75023*w)) - 0.304951*w - 1.07184*cos(0.230234*w)*sin(0.112242 + 0.0138401*x)			
Model: y = -a*x/(b*w + x*w - c) - d*w - e*cos(f*w)*sin(g + h*x) Text: y = -0.372967*x/(2.11815*w + x*w - 0.0956009) - 0.305251*w - 1.06933*cos(0.230186*w)*sin(0.11412 + 0.0138555*x)			

Model	Observed *vs.* Predicted	Output *vs.* Row	Error/Complexity Pareto
Model: y = -a*x/(b*w + x*w) - c*w - d*cos(e*w)*sin(f + g*x) Text: y = -0.372932*x/(2.30444*w + x*w) - 0.305251*w - 1.069*cos(0.23019*w)*sin(0.112471 + 0.0138632*x)			
Model: y = -a*x/(x*w - b) - c*w - d*cos(e*w)*sin(f + g*x) Text: y = -0.347421*x/(x*w - 1.08264) - 0.305689*w - 1.0642*cos(0.230247*w)*sin(0.127317 + 0.0137664*x)			
Model: y = a/(b - c*w) - d*w - e*cos(f*w)*sin(g + h*x) Text: y = 0.0833903/(0.167855 - 0.263315*w) - 0.307905*w - 1.04126*cos(0.229181*w)*sin(0.0834108 + 0.0141917*x)			
Model: y = -a/(w - b) - c*w - d*cos(e*w)*sin(f + g*x) Text: y = -0.448645/(w - 0.921867) - 0.305076*w - 1.04774*cos(0.229841*w)*sin(0.119473 + 0.0138623*x)			
Model: y = a - b*w - c*cos(d*w)*cos(e + f*x - g*w) Text: y = 0.169999 - 0.288049*w - 1.07602*cos(0.23035*w)*cos(4.77206 + 0.0136656*x - 0.0160478*w)			

Model	Observed *vs.* Predicted	Output *vs.* Row	Error/Complexity Pareto
Model: y = -a/w - b*w - c*cos(d*w)*sin(e + f*x) Text: y = -0.110984/w - 0.312571*w - 1.04536*cos(0.228251*w)*sin(0.0167447 + 0.0145702*x)			
Model: y = a - b*w - c*cos(d*w)*cos(e + f*x) Text: y = 0.169991 - 0.287994*w - 1.07427*cos(0.230425*w)*cos(4.80185 + 0.0137989*x)			
Model: y = a - b*w - c*sin(d*x)*cos(e*w) Text: y = 0.0618919 - 0.30506*w - 1.0643*sin(0.0145879*x)*cos(0.229141*w)			
Model: y = a - sin(b*x)*cos(c*w) - d*w Text: y = 0.0517325 - sin(0.0148312*x)*cos(0.22667*w) - 0.307702*w			
Model: y = sin(-a*x)*cos(-b*w) - c*w Text: y = sin(-0.0148891*x)*cos(-0.223831*w) - 0.31763*w			

Model	Observed *vs.* Predicted	Output *vs.* Row	Error/Complexity Pareto
Model: y = a + x2/(b + x2 - c*x) Text: y = 0.477783 + x^2/(96163 + x^2 - 398.901*x)			
Model: y = -a*sin(-b*x)/w - c*w Text: y = -2.15477*sin(-0.0153494*x)/w - 0.327939*w			
Model: y = a + b*x - c*cos(d*x) Text: y = 0.945533 + 0.00177828*x - 0.49458*cos(0.00752338*x)			
Model: y = a*sin(-b*x) - c*w Text: y = 0.624069*sin(-0.0157107*x) - 0.311543*w			
Model: y = a + b*x - c*x2 Text: y = 0.393479 + 0.00595572*x - 4.4983e-6*x^2			

Model	Observed *vs.* Predicted	Output *vs.* Row	Error/Complexity Pareto
Model: y = a*sin(b + c*x) Text: y = 2.44793*sin(0.172041 + 0.00214494*x)			
Model: y = a - cos(-b*x) Text: y = 1.47901 - cos(-0.00575565*x)			
Model: y = a + b*w2 Text: y = 0.347961 + 0.0385729*w^2			
Model: y = -a - b*w Text: y = -0.1258 - 0.330803*w			
Model: y = -a*w Text: y = -0.311868*w			



REFERENCES

[1] T.C. Goh, "Nonlinear modelling in geotechnical engineering using neural networks", *Austr. Civil Eng. Trans.*, CE36, vol. 4, pp. 293-297, 1994.

[2] B. Robert and V. Henk, "Neuro-Fuzzy methods for nonlinear system identification", *Ann. Rev. Control*, vol. 27, pp. 73-85, 2003.

[3] S. Horikawa, T. Furuhashi, and Y. Uchikawa, "A neural fuzzy system with fuzzy supervised learning", *IEEE Trans. Sys., Man, Cybern., Pt. B: Cybern.*, vol. 26, no. 5, pp. 244-763, 1996.

[4] C-H. Chen, C-J. Lin, and C-T. Lin, "A recurrent functional-link-based Neural fuzzy system and its applications", *IEEE Sympo. Digital Object Ident.*, pp. 415-420, 2007.

[5] Ö. Ciftcioglu, M.S. Bittermann, and I.S. Sariyildiz, "A neural fuzzy system for soft computing", *In: Proc. 26th Annual Meeting of the North American Fuzzy Information Processing Society - NAFIPS' 07, San Diego, USA*, pp. 489-495, 2007.

[6] L. Cheng, C. Cheng, and L. Chin, "Efficient self-evolving learning for neurofuzzy inference systems", *IEEE Trans. On Fuzzy Systems*, vol. 16, no. 6, pp. 1476-1490, 2008.

[7] E.A. Lim, and S. Yogan, "A study of neuro-fuzzy system in approximation-based problems", *MATEMATIKA*, vol. 24, no. 2, pp. 113-130, 2008.

[8] M.A. Shahin, H.R. Maier, and M.B. Jaksa, "Settlement prediction of shallow foundations on granular soils using B-spline neurofuzzy models", *Comp. Geotech.*, vol. 30, pp. 637-647, 2003.

[9] Z-J. Chen, N-N. Zhang, and X-W. Zhang, "Settlement monitoring system of pile-group foundation", *J. Cent. South Univ. Technol.*, vol. 18, pp. 2122-2130, 2011.

[10] R. Gao, N. Hu, and B. Zhu, "Experimental study and numerical analysis on bearing behaviors of super-long rock-socketed bored pile groups", *J. Southeast Univ.*, 26, vol. 4, pp. 597-602, 2010.

[11] K. Danno and M. Kimura "Evaluation of long-term displacements of pile foundation using coupled fem and centrifuge model test", *Soil Found.*, vol. 49, no. 6, pp. 941-958, 2009.

[12] M.F. Randolph, and C.P. Worth, "Analysis of deformation of vertically loaded piles", *J. Geotech. Engin., ASCE*, 104(GT12), pp. 1465-1488, 1978.

[13] H.G. Poulos, and E.H. Davis, "Pile Foundation Analysis and Design", John Wiley: New york, 1980.

[14] S. Desai, "Numerical design-analysis for piles in sands", *ASCE, 100(GT6)*, pp. 613-635, 1974.

[15] X. Li, Z. Chen, C. Jie, and H. Ruan, "Study on settlement of pile foundation of Sutong Bridge", *In: Proc. Conf. Elect. Technol. Civil Eng., ICETCE*, pp. 772-775, 2011.

[16] Y. Min, "Study on reducing- settlement pile foundation based on controlling settlement principle", *Chinese J. Geotech. Eng.*, vol. 22, no. 4, 2000.

[17] Republic of China Ministry of Railways, "Code for design on subsoil and foundation of railway bridge and culvert", pp. 25-102, 2005.

[18] Republic of China Ministry of Railways, "A total of 200 kilometer per hour passenger railway interim design provisions code", 2007.

[19] AASHTO, "AASHTO-LRFD Bridge Design Specifications", 4th ed., Customary US Units, Washington, DC., pp. 204-209, 2007.

[20] Y-J. Yoon and H-Y. Jang, "SLC. Long-term Settlement Prediction of SUDOKWON Landfill site", Gwon Seoung-Hoon, Tescom Engineering co., 2010.

[21] L.D. Delwiche, and S.J. Slaughter, *The Little SAS Book*: A Primer, 4th Ed. Cary, NC: SAS Institute Inc. 2008.

Received: January 09, 2014 Revised: May 06, 2014 Accepted: May 07, 2014

© Hussein Y. Aziz; Licensee *Bentham Open*.

This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0/) which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.