HYPERMEDIA WEB SOFTWARE EFFORT ESTIMATE WITH ADAPTIVE NEURO FUZZY INFERENCE SYSTEM

MOHAMMAD SABER IRAJI
Faculty Member of Department of Computer Engineering and Information Technology, Payame Noor University, I.R. of Iran
E_mail: iraji.ms@gmail.com

ABSTRACT

Accurate software cost estimates are an important factor in the stability of the software companies in the world competitive and efficient use of resources. Nature and structure of web applications is quite different from traditional software. In 2003, The estimated cost of hypermedia web projects was based on seven features were obtained best results, using case base reasoning (CBR) using Stepwise Regression approaches with MMRE on 37 web hypermedia projects. We considered count of html, count of media files and count of inner links features, presented in this paper proposed approach to reduce predicted effort Error than the actual amount for web hypermedia projects and calculate average relative deviations (AAD), through adaptive neuro fuzzy system (ANFIS) Method that is achieved better and more accurate results.

Keywords: Hypermedia Web Software, Fuzzy, Effort, Cost, Neural Network

1. INTRODUCTION

In the early days of working with computer, software costs made up a small percentage of the total cost of computer systems and So the error in estimating the cost of software, Had a relatively little impact. Today software is the most expensive component of any computer system. For example, in custom systems, a large error in the estimated effort costs can be as the difference between profit and loss. Excessive costs would be catastrophic for the software manufacturer.

Humans always look forward to reducing costs and increase quality. Today software cost is very important in a software engineering field of computer science. Software effort estimate is very complex problem and is Software engineering consideration. Improvement exact prediction effort enables software engineering managers to manage the software project better.

In the past, application was traditional Windows-based, but many of today's software companies prefer to produce their web application. Web app was as a powerful communication interface for communicating with customers and markets. Due to easy access and comprehensive web. Web applications are classified in tow category web hypermedia involve non

programming languages (html, media) and web software application with programming languages (java, dcom, activex)[4].

Development of web has led to the web engineering definition and is created useful web applications with the Internet. Therefore, requires an estimate of the cost model for Web applications is essential.

Web-based software costing is very difficult, some of the reasons below, we count on:

1- The use of high-skilled graphic design, and programming is down [1].
2- Problem exists for web sizing because the use of different programming languages such as java, html, xml, activex [2].
3- Producing high-quality web applications up to six months [3].

Determine the size of software is commonly used to predict software effort. Many techniques have been developed to estimate the costs, which they refer below.

This paper is extracted from a project funded by the Payame Noor University has been adopted as “hypermedia web software effort estimate with adaptive neuro fuzzy inference system”.

133
1.1 Algorithmic Models

Parameter mathematical model, using the analysis of data from old projects is calibrated together. This type of model provides a simple relationship between the software project effort and other characteristics such as size (loc, fp, ufp, usp,...) example: the Constructive Cost Model (COCOMO), and Classification and Regression Trees (CART)[5], ordinary least-squares regression (OLS)[6].

Web object was introduced by Reifer[7] for web application sizing as function points. Web objects, including elements of (1) internal logical files; (2) external interface file; (3) external input; (4) external output; (5) external inquiries; (6) multimedia files; (7) web building blocks; (8) scripts; and (9) links [8]. Ruhe, Jeffery and Wieczorek use web objects for estimate web application effort [9].

Mangia and Paiano were suggested Metric Model for Web Application (MMWA) metrics with calculate complexity factors in development web application. MMWA is sub- Split into four subcategories recognized as (i) Functional Sizing Model; (ii) Navigational Structures Sizing Model; (iii) Publishing Sizing Model; and (iv) Multimedia Sizing Model [10].

1.2 Machine Learning

Machine learning approach uses Genetic Algorithms, neural networks, fuzzy logic, case based reasoning (CBR) methods and can automatically discover patterns in the training data, and software cost estimate. A genetic algorithm approach used objective function to optimize effort problem with big populations [13]. Neural networks like human brain learn to predict software effort through neurons. A fuzzy system is a relationship between the input and output based on linguistic variables and rules of inference. In [11] was used use case size point method for Object Oriented Software Effort Estimate with Adaptive Neuro Fuzzy and resulted in effort determination a more accurate. Case based Reasoning (CBR) is a method where information is used to solve new cases from past cases for a cost estimate by similarity function (The similarity rate as Euclidean distance) and analogy adaptation scheme [14].

1.3 Expert Judgment

This method is more experts predictions based on their skills and experiments. Ruhe, et all suggested a method based two elements recognized as 1-causal model (cost driver), 2-data from last projects to web cost estimation by researching into application of COBRA (Cost Estimation, Benchmarking, and Risk Assessment)[12].in COBRA, models are used, the experience of experts in order to determine the qualitative and quantitative factors for overhead costs.

The paper is organized in five sections. After the introduction Section 1, Section 2 which also introduces the related works' hypermedia web software cost estimate Section 2 continues with Mathematical Model in section 3. Section 4 and 5 presents the results, conclusions of the research. The paper ends with a list of references.

2. LITERATURE

Ruhe, et al Offered Ordinary least-squares regression method with Magnitude relative error (MRE) On Industrial Australian web development company (12 data set) by Web objects,(WebMo),Vs , Function point's Size and best result were obtained for Web Objects[9].

(Costagliola et al. 2006) Described Linear Regression(LR),Regression tree(RT),Stepwise regression(SW),Analogy-based estimation(ABE), Combination of RT and LR, Combination of RT and ABE methods with MMRE, MdMRE on Italian software company (15 web projects) by Length measures(number of web pages, new web pages, scripts, link references), Functional Measures (Fp+web objects) and Their results showed that the method LR – RT has a better result[15].

Mendes et al. suggested Stepwise Regression, Regression Trees approached with MMRE, MdMRE on 37 web hypermedia projects developed by Msc Students University of Auckland by Page's count, Media Count, Program Count, Connectivity density, total page complexity, reused media count, reused program count and Better results were obtained using Stepwise Regression [16].

Mendes et al. recommended Bayesian network, Manual stepwise regression methods with MMRE, MdMRE on 150 web application projects from Tukutuku database, 25 variables that Bayesian networks were better [17].
Idris et.al introduced Fuzzy Radial Basis Function Neural networks (FRBFN) methods on 53 web hypermedia from Tukutuku dataset, nine numerical attributes with MMRE index and concluded that the results an RBFN using fuzzy C-means achieved better than RBFN using hard C-means [18].

Corazza et al. proposed Support Vector Regression (SVR), Manual Stepwise Regression, Bayesian networks approaches on 130 web application projects unsystematic selected from Tukutuku database. That the best results were obtained by Support Vector Regression (SVR) [19].

Corazza et al. used the meta-heuristics Tabu Search (TS) for the sake of automatically select suitable SVR parameters with RBF kernel function on single and cross-company 21 data sets and obtained a more appropriate result [20].

In the literature review above, try to minimize the error between the predicted costs with the actual costs of the web application based on input characteristics from the different database to estimate the cost of web projects.

In this paper, we first said about a cost estimates for web software and need it. Then we pay attention to many papers in this field that have been considered paper [16] data as the base of the research and by neural network and ANFIS method, we are trying to estimate more accurately the cost of a web hypermedia software.

3. METHODOLOGY AND MODEL

Soft computing are included, different types of neural networks, fuzzy systems, genetic algorithms, etc. That in information retrieval applications. Fuzzy theory was developed by Zadeh [22], a new intelligent method stated to solve Unlike problems than the old calculations.

Neural networks have made of neurons and used for modelling between input and output. The middle(hidden) layer is responsible for communication. Neural network output calculates from bellow equation.

\[
o^h_{j} = f^h_j(\sum^n_{i=1} w^h_{ij} p_i + b^h_j)
\]

\[
o^h_{j} = \text{output of neuron } j \text{ of hidden neuron}
\]

\[
p_i = \text{input } i \text{ to hidden neuron}
\]

\[
w^h_{ij} = \text{weight connection between input and hidden neuron from input } i \text{ to neuron } j
\]

\[
b^h_j = \text{bias hidden neuron } j
\]

\[
f^h_j = \text{transfer function for hidden neuron } j
\]

Transfer function is defined as follows:

\[
\text{purelin}(n) = n
\]

\[
\text{tansig}(n) = \frac{2}{(1 + \exp(-2n))} - 1
\]

We used feed forward neural network and BP (Back propagation) learning method in this research (Figure 1). For further information about neural network, refer to [24].

Neural networks use the data to predict that 70% of the randomly selected for train data, 15% for validation and test.

ANFIS (Adaptive Neuro Fuzzy Inference System) is based sugeno (Jang, Sun & Mizutani, 1997; Jang & Sun, 1995) A generic rule in a Sugeno fuzzy pattern has the form If Input 1 = x and Input 2 = y, then output is z = ax + by + c. Figure 2 explains the ANFIS neural network [23].
Figure 2. Adaptive Neuro-fuzzy Network (ANFIS) for Web Hypermedia Software Cost

In Figure 2 first layer are the degree of membership of linguistic variables. The second layer is 'rules layer'. After the linear composition of rules at third layer then specify the degree of belonging to a special class by Sigmoid's function in layer 4. ANFIS is a type of fuzzy neural network with a learning algorithm based on a set of training data for tuning an available rule base that permits the rule base to reconcile the training data.

Total-effort be calculated from the following equation:

\[
\text{Total effort} = \sum_{i=1}^{n} \text{PAE} + \sum_{j=0}^{m} \text{MAE} + \sum_{k=0}^{p} \text{PRE}
\]

Where PAE is the page authoring effort, MAE the media authoring effort and PRE the program authoring effort [21].

We have applied Page count, Media Count, Connectivity density, total page complexity, reused media count, entries to ANFIS the given training data. The related rules is set, and obtain more accurate output (Figure 2). These features are defined below.

1. Page count (PaC) count of html or shtml files used in the application.
2. Media count (MeC) count of media files used in the application.
3. Reused media count (RMC) count of reused/modified media files.
4. Connectivity density (COD) Total count of inner links divided by Page Count.
5. Total page complexity (TPC) Average count of various types of media per page.
6. Total effort (TE) Effort in person hours to design and build the application.

4. EXPERIMENTAL RESULTS

We implement our proposed system in MATLAB version 7.12 on Laptop, 1.7 GHZ CPU, used the absolute average percent deviation (AAD %) and root mean square error (RMS) indicators.
In order to determine the number of hidden layer neurons,

\[ AAD\% = \frac{1}{N} \sum_{i=1}^{N} 100 \times \left| \frac{\text{effort}_{\text{actual}} - \text{effort}_{\text{predicted}}}{\text{effort}_{\text{actual}}} \right| \] (4)

\[ \text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (\text{effort}_{\text{actual}} - \text{effort}_{\text{predicted}})^2 \] (5)

\[ \text{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\text{effort}_{\text{actual}} - \text{effort}_{\text{predicted}})^2} \] (6)

Figure 3. (a) RMSE and (b) AAD Between Actual Effort and Predicted Effort Using Neural Network for Determine Number of Hidden Neuron

Figure 3 shows RMSE and AAD between actual effort and predicted effort using neural network from 1 to 15 hidden neuron. As seen in Figure, the optimum number of hidden neurons is 12.

Neural network architecture 5-12-1 were considered and reached its best performance after three epochs.

Figure 4 shows the amount of MSE for the train validation, test data using best topology of neural network. For the best network performance at epochs three Mse for train, validation and test data is 0.8806, 61.9108 and 758.1232 In Figure 4.

Figure 4. (a) Training Validation and Test Error Curve (b) best Performance with Neural Network for Validation Data is Epochs 3

Figure 5 shows the correlation coefficient, accordance the predicted effort and actual effort namely \( R^2 \) for the train, validation, test and total data.

In ANFIS proposed system was considered a database of 34 web projects, In order to train and test the fuzzy neural network. After calculate 6 Features Page count, Media Count, Connectivity
density, total page complexity, reused media count, described above for 34 web projects, 26 web page was considered for train ANFIS and 8 web projects was allocated to test system.

After setting network parameters to generate fis = grid partition, optim.method=hybrid, linier, train fis epochs=8, gaussmf membership function with 2 mf Rmse (Root mean squared errors) for training data obtained 0.0000.

Figure 6 shows the deal of MSE for the train validation, test data using neural network and ANFIS. As is evident MSE for the best ANFIS system performance at epochs 4 train, validation and test data is 0.0000, 168.3865 and 198.4078.

Figure 5. Plot Predicted Effort and Actual Effort $R^2$ for (a) Train (b) Validation (c) Test (d) Total data

Correlation coefficient demonstrates predicted effort and actual effort namely $R^2$ for the train, validation, test and total data in Figure 7. The $R^2$ are compared in Table 1 using neural network and ANFIS proposed system. The $R^2$ for train data use ANFIS system is 1 that proving the ANFIS system is better performance in the training phase than neural network. Finally compared to $R^2$ total data, suggest that the ANFIS system is ultimately best performance with $R^2$ =0.9689 versus $R^2$ =0.9179 using neural network.

Table 1. Comparison $R^2$ using neural network and ANFIS

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Validation</th>
<th>Test</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neural network</td>
<td>0.9993</td>
<td>0.9759</td>
<td>0.7745</td>
<td>0.9179</td>
</tr>
<tr>
<td>ANFIS</td>
<td>1</td>
<td>0.928</td>
<td>0.9321</td>
<td>0.9689</td>
</tr>
</tbody>
</table>

Figure 8 shows deviations of predicted effort from actual effort (Dev %) for hypermedia web
projects. Values of deviations for predicted effort of hypermedia web projects have been presented in table 2 (Msc Students University of Auckland data).

Table 3, 4 show AAD, RMSE, minimum, maximum deviation using ANFIS and neural network proposed system. The average relative deviations for train, validation, test and complete data are 0, 0.6846, 6.6425, 16.389 and 3.8702 using neural network respectively. According to tables 3, 4 the root mean square error for train, validation , test and total data are 0, 0.9384, 7.8683, 27.534 and 11.0098 using neural network. The results show that the ANFIS model can accurately estimate effort for web hypermedia with high accuracy.

Pred(p) factor is defined in equation 5. Where N is the total number of observations, k is the number of observations with a DEV less than or equal to p. A common value for p is 25.

\[ \text{pred}(p) = \frac{k}{N} \]  

The pred (25) and AAD for total data using ANFIS model presented in this work is 100 and 2.4909 versus model presented in references [16].
5. CONCLUSION

Our purpose in this research is to predict the exact quantity of web hypermedia software effort and creating two models: neural network model and adaptive neuro-fuzzy model for this purpose. A neural network model with (5-13-1) structure and ANFIS model with five inputs and gaussmf membership function with two mf were designed to predict web effort. From the total number of data, 75% were randomly chosen for train, 15% for validation and 15% for test.

Our results show that the amount of the average relative deviations (AAD) indicator with ANFIS is lesser than the existing proposed algorithms of expression in Literature Review among CBR by Mendes with pred(25)=100% and the performed calculations in experimental results in tables 1, 2, 3 prove this claim. Average relative deviations (AAD) for train and total data are 0 and 2.4909 using ANFIS.

In order to perform future works, the proposed model for web application software’s can be developed with raising the data relating to projects, and also other's neural methods can be used in order to determine the exact amount of effort in industrial environments and other data sets to be achieved better results probably by changing the number of linguistic variables, the type of membership function.

<table>
<thead>
<tr>
<th>project#</th>
<th>page count</th>
<th>media count</th>
<th>connectivity density</th>
<th>total page complexity</th>
<th>reused media count</th>
<th>actual effort</th>
<th>nn effort</th>
<th>ANFIS effort</th>
<th>Relative dev(nn)%</th>
<th>Relative dev(ANFIS)%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>43</td>
<td>0</td>
<td>8.72</td>
<td>1.18</td>
<td>42</td>
<td>79.13</td>
<td>79.44519</td>
<td>79.130003</td>
<td>-0.776594</td>
<td>-0.000004</td>
</tr>
<tr>
<td>2</td>
<td>75</td>
<td>21</td>
<td>16.85</td>
<td>1</td>
<td>64</td>
<td>145.5</td>
<td>144.986479</td>
<td>145.499976</td>
<td>0.352936</td>
<td>0.000016</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>2</td>
<td>9.02</td>
<td>1</td>
<td>1</td>
<td>135.4</td>
<td>87.576307</td>
<td>147.211222</td>
<td>35.320305</td>
<td>-8.723206</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td>82</td>
<td>13.9</td>
<td>1</td>
<td>27</td>
<td>128.4</td>
<td>128.33211</td>
<td>128.399982</td>
<td>0.052857</td>
<td>0.000014</td>
</tr>
<tr>
<td>5</td>
<td>53</td>
<td>11</td>
<td>7.58</td>
<td>1</td>
<td>57</td>
<td>106.6</td>
<td>106.305255</td>
<td>104.089334</td>
<td>0.276497</td>
<td>2.355222</td>
</tr>
<tr>
<td>6</td>
<td>52</td>
<td>36</td>
<td>6.37</td>
<td>1.07</td>
<td>43</td>
<td>112.6</td>
<td>113.65535</td>
<td>116.070231</td>
<td>-0.937225</td>
<td>-3.081910</td>
</tr>
<tr>
<td>7</td>
<td>50</td>
<td>13</td>
<td>10.62</td>
<td>1</td>
<td>8</td>
<td>87.05</td>
<td>87.280622</td>
<td>87.049999</td>
<td>-0.26497</td>
<td>0.000001</td>
</tr>
<tr>
<td>8</td>
<td>60</td>
<td>0</td>
<td>21.43</td>
<td>0.28</td>
<td>2</td>
<td>81.54</td>
<td>83.607543</td>
<td>81.539998</td>
<td>-2.535618</td>
<td>0.000002</td>
</tr>
<tr>
<td>9</td>
<td>51</td>
<td>0</td>
<td>7.2</td>
<td>1</td>
<td>89</td>
<td>113.8</td>
<td>115.270943</td>
<td>113.799988</td>
<td>-1.292569</td>
<td>0.000010</td>
</tr>
<tr>
<td>10</td>
<td>51</td>
<td>74</td>
<td>21.63</td>
<td>1.94</td>
<td>75</td>
<td>153.8</td>
<td>162.043916</td>
<td>181.644795</td>
<td>-5.360153</td>
<td>-18.104548</td>
</tr>
<tr>
<td>11</td>
<td>61</td>
<td>8</td>
<td>9</td>
<td>2.07</td>
<td>50</td>
<td>112.1</td>
<td>112.107441</td>
<td>112.000006</td>
<td>-0.095930</td>
<td>-0.000005</td>
</tr>
<tr>
<td>12</td>
<td>66</td>
<td>0</td>
<td>2.58</td>
<td>0.77</td>
<td>66</td>
<td>122.2</td>
<td>121.500213</td>
<td>122.200001</td>
<td>0.572658</td>
<td>-0.000001</td>
</tr>
<tr>
<td>13</td>
<td>59</td>
<td>66</td>
<td>16.54</td>
<td>1.88</td>
<td>15</td>
<td>125.1</td>
<td>124.368682</td>
<td>125.100005</td>
<td>0.584587</td>
<td>-0.000004</td>
</tr>
<tr>
<td>14</td>
<td>59</td>
<td>13</td>
<td>15.53</td>
<td>2.51</td>
<td>82</td>
<td>128.5</td>
<td>128.624536</td>
<td>128.500000</td>
<td>-0.096915</td>
<td>0.000000</td>
</tr>
<tr>
<td>15</td>
<td>50</td>
<td>5</td>
<td>12.24</td>
<td>1</td>
<td>81</td>
<td>115.5</td>
<td>111.459034</td>
<td>101.128308</td>
<td>3.498672</td>
<td>12.443024</td>
</tr>
<tr>
<td>16</td>
<td>53</td>
<td>63</td>
<td>23.3</td>
<td>1.11</td>
<td>7</td>
<td>119.7</td>
<td>119.526741</td>
<td>119.700000</td>
<td>0.144745</td>
<td>0.000000</td>
</tr>
<tr>
<td>17</td>
<td>53</td>
<td>30</td>
<td>1.7</td>
<td>0.17</td>
<td>10</td>
<td>106.1</td>
<td>69.229601</td>
<td>109.454397</td>
<td>34.750612</td>
<td>-3.161543</td>
</tr>
<tr>
<td>18</td>
<td>55</td>
<td>0</td>
<td>6.84</td>
<td>0</td>
<td>0</td>
<td>73.81</td>
<td>71.84855</td>
<td>73.810001</td>
<td>2.662438</td>
<td>-0.000002</td>
</tr>
<tr>
<td>19</td>
<td>44</td>
<td>126</td>
<td>13.95</td>
<td>1</td>
<td>30</td>
<td>147.4</td>
<td>147.303933</td>
<td>147.400005</td>
<td>0.065174</td>
<td>-0.000003</td>
</tr>
<tr>
<td>20</td>
<td>66</td>
<td>27</td>
<td>13.58</td>
<td>1</td>
<td>31</td>
<td>120.9</td>
<td>119.447690</td>
<td>120.000000</td>
<td>0.460259</td>
<td>0.000000</td>
</tr>
<tr>
<td>21</td>
<td>43</td>
<td>0</td>
<td>8.72</td>
<td>1.19</td>
<td>30</td>
<td>73.01</td>
<td>78.569714</td>
<td>75.120575</td>
<td>-7.61503</td>
<td>-2.890802</td>
</tr>
<tr>
<td>22</td>
<td>56</td>
<td>25</td>
<td>2.77</td>
<td>1.75</td>
<td>15</td>
<td>97.3</td>
<td>96.714891</td>
<td>97.300002</td>
<td>0.601345</td>
<td>-0.000002</td>
</tr>
<tr>
<td>23</td>
<td>53</td>
<td>0</td>
<td>4.87</td>
<td>0.19</td>
<td>10</td>
<td>76.230000</td>
<td>74.794572</td>
<td>76.230000</td>
<td>1.883023</td>
<td>0.000000</td>
</tr>
<tr>
<td>24</td>
<td>51</td>
<td>20</td>
<td>16.67</td>
<td>1</td>
<td>112</td>
<td>137.2</td>
<td>139.347305</td>
<td>137.199977</td>
<td>-1.565091</td>
<td>0.000017</td>
</tr>
<tr>
<td>25</td>
<td>55</td>
<td>25</td>
<td>4.33</td>
<td>1</td>
<td>57</td>
<td>117.4</td>
<td>116.955634</td>
<td>117.400000</td>
<td>0.378306</td>
<td>0.000000</td>
</tr>
<tr>
<td>26</td>
<td>52</td>
<td>48</td>
<td>16.31</td>
<td>1.85</td>
<td>45</td>
<td>141.14</td>
<td>140.922428</td>
<td>141.399997</td>
<td>0.337746</td>
<td>0.000002</td>
</tr>
</tbody>
</table>
Table 3. AAD, RMSE, Minimum, Maximum deviation using ANFIS

<table>
<thead>
<tr>
<th>ANFIS</th>
<th>train</th>
<th>validation</th>
<th>test</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAD%</td>
<td>0</td>
<td>9.3645</td>
<td>7.5738</td>
<td>2.4909</td>
</tr>
<tr>
<td>RMSE</td>
<td>0</td>
<td>12.9764</td>
<td>14.0857</td>
<td>7.3444</td>
</tr>
<tr>
<td>Dev min(%)</td>
<td>0.0000</td>
<td>-18.0868</td>
<td>-18.1045</td>
<td>-18.104548</td>
</tr>
<tr>
<td>Dev max(%)</td>
<td>0.0000</td>
<td>12.4430</td>
<td>5.5244</td>
<td>12.4430</td>
</tr>
<tr>
<td>No of Dev&lt;0.25</td>
<td>24</td>
<td>5</td>
<td>5</td>
<td>34</td>
</tr>
</tbody>
</table>

Table 4. AAD, RMSE, Minimum, Maximum deviation using NN

<table>
<thead>
<tr>
<th>Neural network</th>
<th>train</th>
<th>validation</th>
<th>test</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAD%</td>
<td>0.6846</td>
<td>6.6425</td>
<td>16.389</td>
<td>3.8702</td>
</tr>
<tr>
<td>RMSE</td>
<td>0.9384</td>
<td>7.8683</td>
<td>27.534</td>
<td>11.0098</td>
</tr>
<tr>
<td>Dev min(%)</td>
<td>-2.53562</td>
<td>-11.271</td>
<td>-5.3602</td>
<td>-11.271</td>
</tr>
<tr>
<td>Dev max(%)</td>
<td>2.6624</td>
<td>3.4987</td>
<td>35.3203</td>
<td>35.3203</td>
</tr>
<tr>
<td>No of Dev&lt;0.25</td>
<td>24</td>
<td>5</td>
<td>3</td>
<td>32</td>
</tr>
</tbody>
</table>

**APPENDIX A. CODE**

Codes for validation inputs using ANFIS system:
y2 = evalfis(valinputs',chk_out_fismat);
ANFISvalinputs2=valinputs'
ANFISvaltargets2=valtargets'
y22=y2
ANFISvaldev=(ANFISvaltargets2 -y22)
ANFISrmseval=sqrt(mse(ANFISvaldev))
ANFISvaldevabs=abs(ANFISvaldev)
rANFISvaldev=ANFISvaldev./ANFISvaltargets2
rANFISvaldev100=rANFISvaldev*100
ANFISvaldevabs2=ANFISvaldevabs./ANFISvaltarget s2
ANFISvaldevabs2=ANFISvaldevabs2*100
[ANFISnval nv]=size(ANFISvaldevabs2)
ANFISaadval100=sum(ANFISvaldevabs2)/ANFISnval

**REFERENCES**


International Symposium on Software Metrics, p.30


