Journal of Theoretical and Applied Information Technology
15th November 2016. Vol.93. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

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".

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

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].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

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 at al. used the meta-heuristics Tabu Search (TS) for the sake 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.

$$o_j^h = f_j^h\left(\sum_i^n w_{ij}^h p_i + b_j^h\right) \quad (1)$$

o_j^h=output of neuron j of hidden neuron
p_i=input i to hidden neuron
w_{ij}^h =weight connection between input and hidden neuron from input i to neuron j
b_j^h=bias hidden neuron j
f_j^h= transfer function for hidden neuron j

transfer function Is defined as follows:
purelin (n)=n

$$tansig(n) = \frac{2}{(1+\exp(-2n)-1} \quad (2)$$

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].

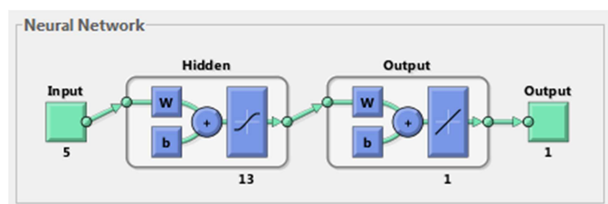


Figure 1. Neural Network Structure

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].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

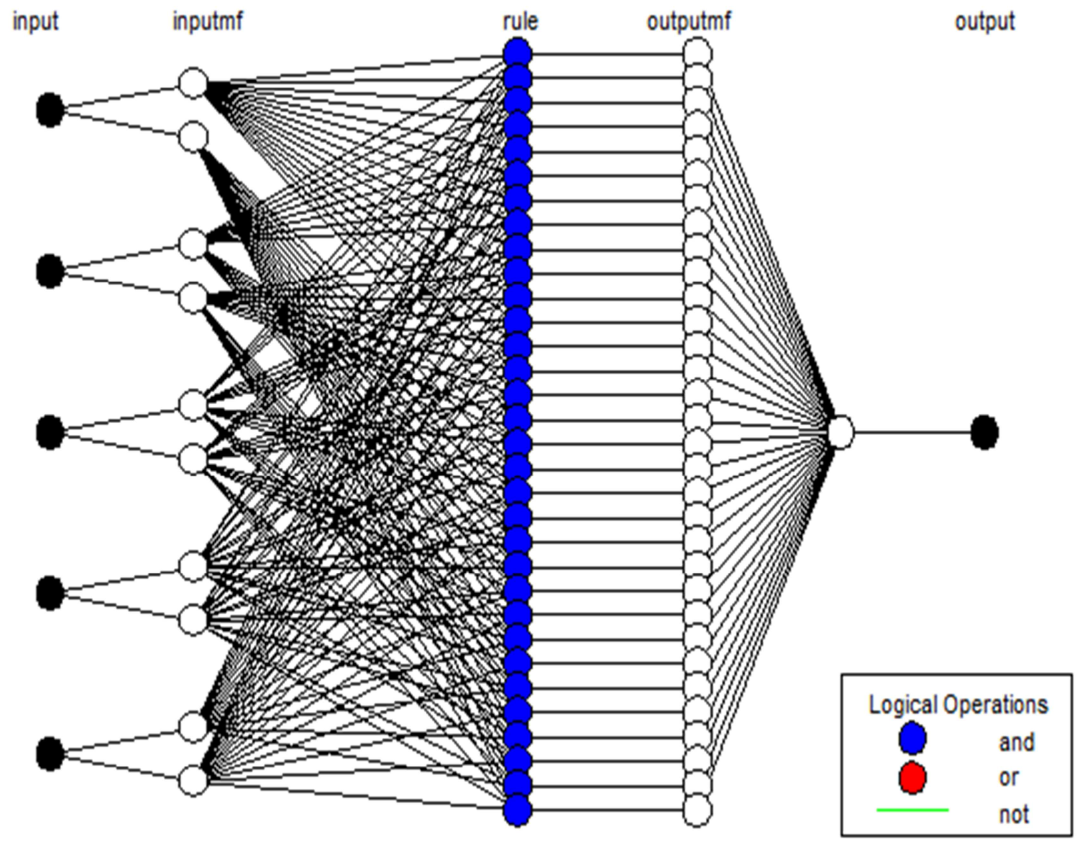


Figure 2. Adaptive Neuro fuzzy Network (ANFIS) for Web Hypermedia Software Cost

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 Sigmund'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:

$$\text{Totaleffort}= \quad \sum_{i=1}^{n} \text{PAE} + \sum_{j=0}^{m} \text{MAE} + \sum_{k=0}^{p} \text{PRE} \quad (3)$$

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. EXPRIMENTAL 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

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In order to determine the number of hidden layer neurons.

$$AAD\% = \frac{1}{N}\sum_{1}^{N} 100 * \left| \frac{effort_{actual} - effort_{predicted}}{effort_{actual}} \right| \quad (4)$$

$$MSE = \frac{1}{N}\sum_{1}^{N}\left(effort_{actual} - effort_{predicted}\right)^2 \quad (5)$$

$$RMS = \sqrt{\frac{1}{N}\sum_{1}^{N}\left(effort_{actual} - effort_{predicted}\right)^2} \quad (6)$$

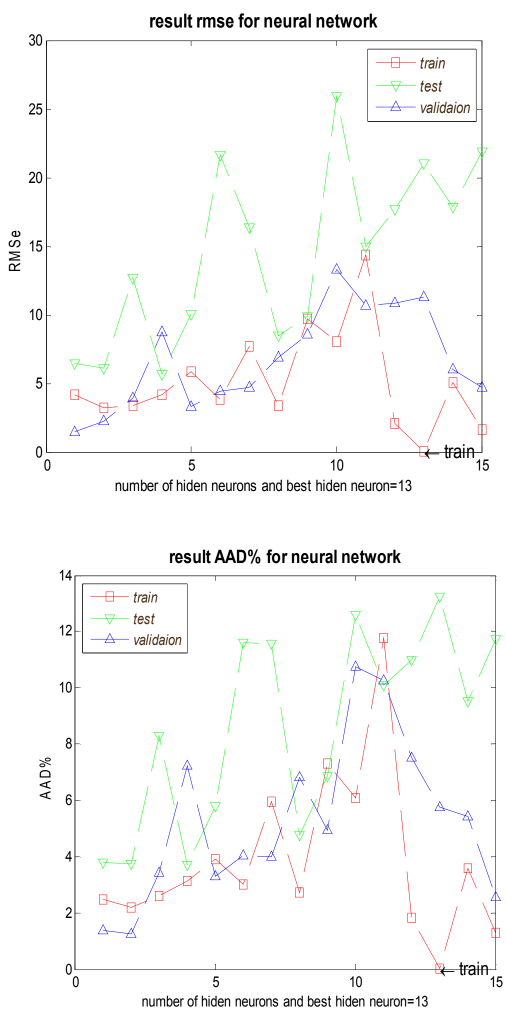


Figure 3. (a) RMSE and (b) AAD Between Actual Effort and Predicted Effort Using Neural Network for Determine Number of Hidden Neuron

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.

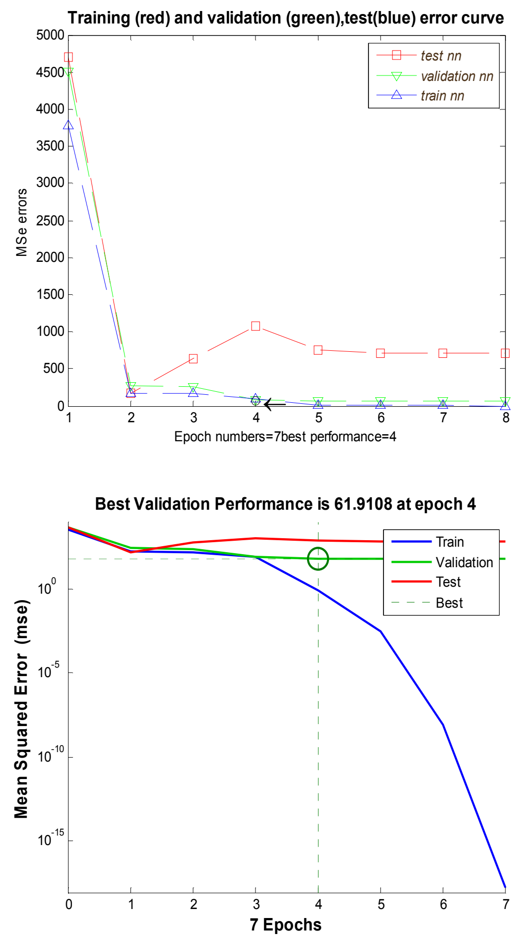


Figure 4.(a) Training Validation and Test Error Curve (b)best Performance with Neural Network for Validation Data is Epochs 3

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

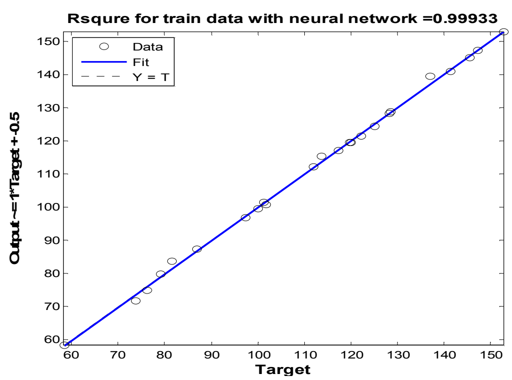
Journal of Theoretical and Applied Information Technology
15th November 2016. Vol.93. No.1

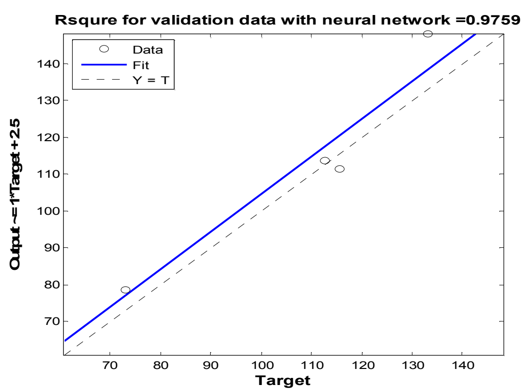
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

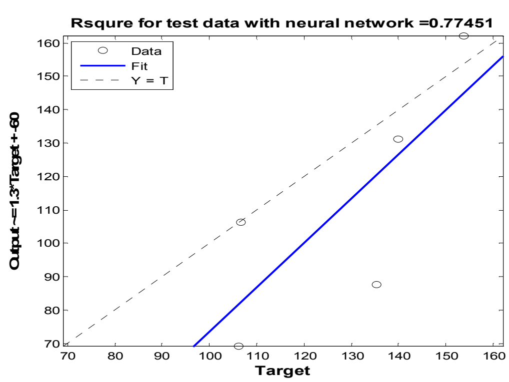
density, total page complexity, reused media count, described above for 34 web projects, 26 web page was considered for train ANFIS and 8 web projects Was Allocated to Test system.

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.







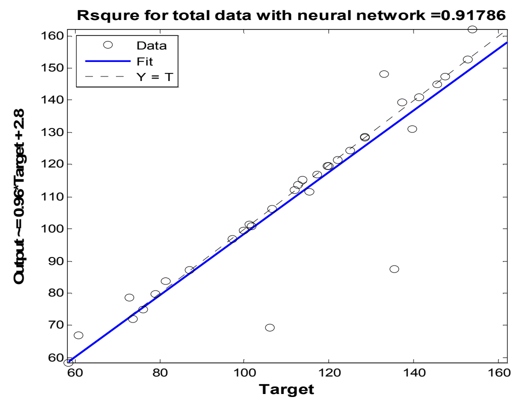


Figure 5. Plot Predicted Effort and Actual Effort R^2 for (a) Train (b) Validation (c) Test (d)Total data

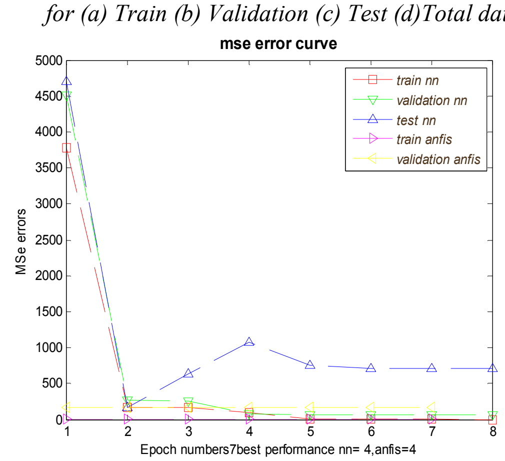


Figure 6. Comparison Training, Validation and Test Error with Neural Network and ANFIS

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 table1 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

R^2	Train	validation	test	Total
Neural network	0.9993	0.9759	0.7745	0.9179
ANFIS	1	0.928	0.9321	0.9689

Figure 8 shows deviations of predicted effort from actual effort (Dev %) for hypermedia web

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

projects. Values of deviations for predicted effort of hypermedia web projects have been presented in table 2 (Msc Students University of Auckland data).

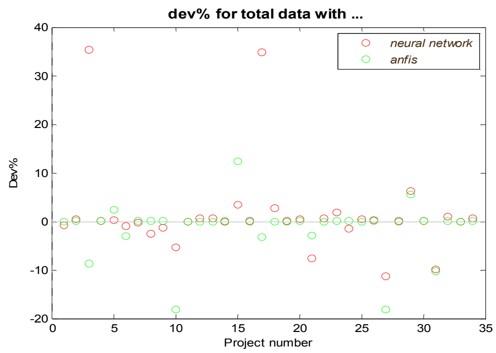
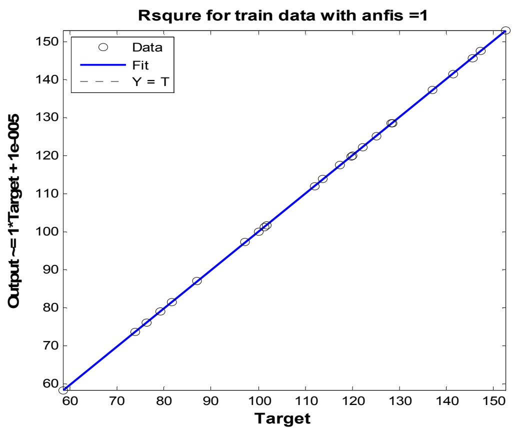
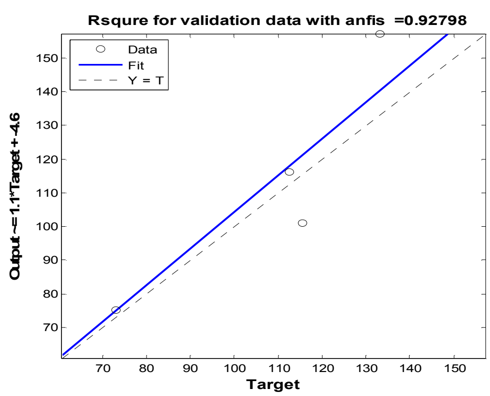
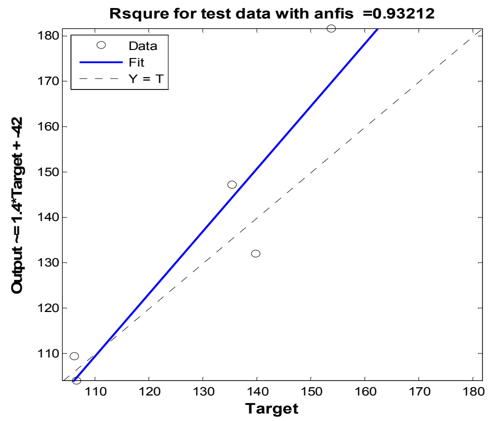


Figure 8. Comparison Predicted Effort by Neural Network and ANFIS







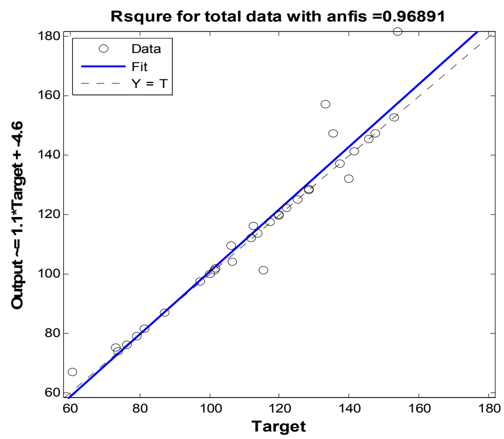


Figure 7. Plot Predicted Effort and Actual Effort R^2 for (a) Train (b) Validation (c) Test (d)Total data using ANFIS system

Tables 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, 9.3645, 7.5738 and 2.4909 using ANFIS compared with 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, 12.9764, 14.0857 and 7.3444 using ANFIS compared with 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.

$$\text{pred}(p) = \frac{k}{N} \qquad (5)$$

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].

Journal of Theoretical and Applied Information Technology
15th November 2016. Vol.93. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5. CONCLUSION

Our purpose in this research is prediction the exact quantity of web hypermedia software effort and creating tow 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 prediction 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 2. Comparison Predicted Effort using Propose Neural Network and ANFIS system

project#	page count	media count	connectivity density	total page complexity	reused media count	actual effort	nn effort	ANFIS effort	Relative dev(nn)%	Relative dev(ANFIS)%
1	43	0	8.72	1.18	42	79.13	79.744519	79.130003	-0.776594	-0.000004
2	75	21	16.85	1	64	145.5	144.986479	145.499976	0.352936	0.000016
3	100	2	9.02	1	0	135.4	87.576307	147.211222	35.320305	-8.723206
4	50	82	13.9	1	27	128.4	128.332131	128.399982	0.052857	0.000014
5	53	11	7.58	1	57	106.6	106.305255	104.089334	0.276497	2.355222
6	52	36	6.37	1.07	43	112.6	113.655315	116.070231	-0.937225	-3.081910
7	50	13	10.62	1	8	87.05	87.280602	87.049999	-0.264907	0.000001
8	60	0	21.43	0.28	2	81.54	83.607543	81.539998	-2.535618	0.000002
9	51	0	7.2	1	89	113.8	115.270943	113.799988	-1.292569	0.000010
10	51	74	21.63	1.94	75	153.8	162.043916	181.644795	-5.360153	-18.104548
11	61	8	9	2.07	50	112	112.107441	112.000006	-0.095930	-0.000005
12	66	0	2.58	0.77	66	122.2	121.500213	122.200001	0.572658	-0.000001
13	59	66	16.54	1.88	15	125.1	124.368682	125.100005	0.584587	-0.000004
14	59	13	15.53	2.51	82	128.5	128.624536	128.500000	-0.096915	0.000000
15	50	5	12.24	1	81	115.5	111.459034	101.128308	3.498672	12.443024
16	53	63	23.3	1.11	7	119.7	119.526741	119.700000	0.144745	0.000000
17	53	30	1.7	0.17	10	106.1	69.229601	109.454397	34.750612	-3.161543
18	55	0	6.84	0	0	73.81	71.844855	73.810001	2.662438	-0.000002
19	44	126	13.95	1	30	147.4	147.303933	147.400005	0.065174	-0.000003
20	66	27	13.58	1	31	120	119.447690	120.000000	0.460259	0.000000
21	43	0	8.72	1.19	30	73.01	78.569714	75.120575	-7.615003	-2.890802
22	56	25	2.77	1.75	15	97.3	96.714891	97.300002	0.601345	-0.000002
23	53	0	4.87	0.19	10	76.23	74.794572	76.230000	1.883023	0.000000
24	51	20	16.67	1	112	137.2	139.347305	137.199977	-1.565091	0.000017
25	55	25	4.33	1	57	117.4	116.955634	117.400000	0.378506	0.000000
26	52	48	16.31	1.85	45	141.4	140.922428	141.399997	0.337746	0.000002

Journal of Theoretical and Applied Information Technology
15th November 2016. Vol.93. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

27	53	53	17.74	2.21	53	133.1	148.101753	157.173527	-11.271039	-18.086797
28	41	4	3.2	1	2	58.36	58.345138	58.360001	0.025466	-0.000001
29	62	21	12.27	1.99	87	139.8	131.080185	132.076919	6.237350	5.524378
30	66	28	7.21	0.98	94	152.8	152.709753	152.799999	0.059062	0.000001
31	33	16	5	1	6	60.79	66.802328	67.063358	-9.890323	-10.319720
32	53	1	1.7	1.05	59	101.8	100.801393	101.800000	0.980950	0.000000
33	54	26	13.2	1.07	27	101.3	101.412125	101.300003	-0.110686	-0.000002
34	54	0	2.57	1.26	54	100	99.468904	100.000000	0.531096	0.000000

Table 3.AAD, RMSE, Minimum, Maximum deviation using ANFIS

ANFIS	train	validation	test	total
AAD%	0	9.3645	7.5738	2.4909
RMSE	0	12.9764	14.0857	7.3444
Dev min(%)	0.0000	-18.0868	-18.1045	-18.104548
Dev max(%)	0.0000	12.4430	5.5244	12.4430
No of Dev<0.25	24	5	5	34

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

Neural network	train	validation	test	total
AAD%	0.6846	6.6425	16.389	3.8702
RMSE	0.9384	7.8683	27.534	11.0098
Dev min(%)	-2.53562	-11.271	-5.3602	-11.271
Dev max(%)	2.6624	3.4987	35.3203	35.3203
No of Dev<0.25	24	5	3	32

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./ANFISvaltargets2
ANFISvaldevabs2=ANFISvaldevabs2*100
[ANFISnval nv]=size(ANFISvaldevabs2)
ANFISaadval100=sum(ANFISvaldevabs2)/ANFISnval
```

REFRENCES

[1] Reifer, D., 2002. Ten deadly risks in Internet and intranet software development. IEEE Software, 19(2), March–April, 12–14.

[2] Mendes, E., Mosley, N., Counsell, S., 2002. Comparison of length,complexity and functionality as size measures for predicting Web design and authoring effort. IEE Proceedings on Software 149 (3),86–92.

[3] Offutt, J., 2002. Quality attributes of Web software applications. IEEE Software 19 (2), 25–32.

[4] Christodoulou, S.P., Zafiris, P.A., Papatheodorou, T.S., 2000.WWW2000: the developer_s view and a practitioner_s approach to Web engineering. In: Proc. 2nd ICSE Workshop Web Eng., pp. 75–92

[5] Boehm, B. 1981. Software Engineering Economics, Prentice – Hall

[6] Briand, L. C., I. Wieczorek. 2002. Software Resource Estimation. Encyclopedia of Software Engineering. Volume 2, P-Z (2nd ed., 2002), Marciniak, John J. (ed.) New York: John Wiley & Sons, pp. 1160-1196.

[7] Reifer, D.J., 2000. Web development: estimating quick-to-market software. Software, IEEE , vol.17, no.6, pp.57-64, Nov/Dec 2000

[8] Sukumar Letchmunan,"Pragmatic Cost Estimation for Web Applications", November 2012

[9] Ruhe Melanie , Ross Jeffery , Isabella Wieczorek. 2003. Using Web Objects for Estimating Software Development Effort for Web Applications, Proceedings of the 9th

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

International Symposium on Software Metrics, p.30

[10]Mangia, Leonardo, and Roberto Paiano. "MMWA: a software sizing model for web applications." In Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on, pp. 53-61. IEEE, 2003.

[11]Iraji, Mohammad Saber, and Homayun Motameni. "Object Oriented Software Effort Estimate with Adaptive Neuro Fuzzy use Case Size Point (ANFUSP)." International Journal of Intelligent Systems and Applications (IJISA) 4, no. 6 (2012): 14.

[12] Ruhe Melanie , Ross Jeffery , Isabella Wieczorek. 2003. Cost estimation for web applications, Proceedings of the 25th International Conference on Software Engineering, 285-294

[13]Burgess, Colin J., and Martin Lefley. 2007. Can genetic programming improve software effort estimation? A comparative evaluation, Information and Software Technology, 43 (14), 863-867

[14] Shepperd, Martin. 2007. Software project economics: a roadmap. Future of Software Engineering (FOSE'07), 304-315

[15]Costagliola, G., Di Martino, S., Ferrucci, F., Gravino, C., Tortora, G., Vitiello, G. 2006. Effort estimation modeling techniques: a case study for web applications. In: ICWE'06. Procs. Intl. Conference on Web Engineering, pp. 9–16

[16]Mendes, E., Ian Watson , Chris Triggs , Nile Mosley, and Steve Counsell. 2003. A Comparative Study of Cost Estimation Models for Web Hypermedia Applications, Empirical Software Engineering, v.8 n.2, p.163-196

[17] Mendes, E.; Mosley, N. 2008b. Bayesian Network Models for Web Effort Prediction: A Comparative Study. IEEE Transactions on Software Engineering, vol.34, no.6, pp.723-737.

[18]Idris Ali, Abdelali Zakrani, Mohamed Elkoutbi and Alain Abran, 2008. Fuzzy Radical Basis function Neural Networks for Web Applications Cost Estimation. IEEE 2008: 576-580

[19]Corazza, A., S. Di Martino, F. Ferrucci, C. Gravino, E. Mendes. 2009. Applying support vector regression for web effort estimation using a cross-company dataset," Empirical Software Engineering and Measurement, International Symposium on, pp.

191-202, 2009 3rd International Symposium on Empirical Software Engineering and Measurement

[20]Corazza, Anna, Sergio Di Martino, Filomena Ferrucci, Carmine Gravino, and Emilia Mendes. 2011. Investigating the use of Support Vector Regression for web effort estimation. Empirical Softw. Engg. 16, 2, 211-243

[21]Mendes, E., Mosley, N., and Counsell, S. 2001b. Web metrics – estimating design and authoring effort. IEEE Multimedia. Special Issue on WebEngineering, (Jan.–Mar.): 50–57.

[22] Zadeh, Lotfi A. "Fuzzy logic= computing with words." Fuzzy Systems, IEEE Transactions on 4, no. 2 (1996): 103-111.

[23]Iraji, Mohammad Saber. "Object Oriented Software Usability Estimate with Adaptive Neuro Fuzzy, Fuzzy Svm." International Journal of Information Engineering and Electronic Business (IJIEEB) 5, no. 1 (2013): 40.

[24] Martin T.. Hagan, Howard B.. Demuth, and Mark Hudson Beale. Neural network design. University of Colorado Bookstore, 2002