International Journal of Scientific & Engineering Research, Volume ʃNȯ(ÚÚÜÌʃNȯ2Ì×ÛÌÔÉÌÙıl ɣhш
ISSN 2229-5518

An evaluation of Fuzzy-based models for Software Cost Prediction

Sanaa Elyassami

Abstract— Software Effort Prediction has long been the most critical task in the management of projects. Although many techniques and models have been proposed over the last decades, accurate software development cost estimation is still a challenging task, especially to overcome the uncertainty and imprecision in estimation. The main objective of this research is to investigate the use of fuzzy decision trees to improve software cost estimation accuracy. In this paper, the Fuzzy C5 model is compared with the Fuzzy ID3 and FID models based on two different parameters; Mean Magnitude of Relative Error (MMRE) and Prediction (Pred). Cocomo'81, Isbsg, and Web datasets are used in the evaluation of the different proposed Fuzzy-based Models. After analyzing the results, it had been found that effort estimation using Fuzzy C5 gives better results compared with the Fuzzy ID3 model and with the FID model.

Index Terms— Decision Tree, Effort Estimation, FID, Fuzzy C5, Fuzzy ID3, Fuzzy Logic, Software project.

1 Introduction

Software effort prediction is defined as the amount of time in human hours needed to design the solution, implement, and test the software. Project cost estimation and project planning are usually carried out together. The cost of development is primarily the cost of the effort involved, so the effort computation is used in both the cost and the project schedule estimate. The estimation of the software effort consists of some specific steps. First step consists on obtaining data from previous projects. Second step consists on generation of estimation models. In addition, the last step consists on checking and validating the model, based on the estimates accuracy.

Predict the amount of effort that will be required to develop the software is crucial for better project control of time and budget. However, produce an accurate estimate of software development costs at the beginning of the project life cycle is a very complex task. Molokken and Jorgensen report that software projects expend between thirty percent and forty percent more effort than is estimated [15]. Improving the estimation accuracy and the techniques to produce better estimates continues to attract considerable research attention. In order to achieve accurate estimates and avoid the overestimates and the underestimates, several cost estimation models have been developed and validated in the last few decades.

The modeling technique used in most software cost estimation models is globally based on a mathematical function such as effort=α×sizeβ, where α represents a productivity coefficient, and β indicates the economies/diseconomies scale coefficient factor. However, other cost estimation models are based on computational intelligence techniques such as case based reasoning [18], decision trees [24], artificial neural networks [10] [23] and fuzzy logic based models [16] [22]. The decision tree method is widely used for inductive learning and has been demonstrating its superiority in terms of predictive accuracy in many fields [2] [17]. The most widely used algorithms for building decision trees are ID3 [19], C4.5 [20] and CART [1].

When building prediction models for software cost estimation, the primary goal should be to make a model that most accurately predicts the desired target estimation for new projects. Several attempts have been made to renew some of the current models, by introducing fuzzy logic, in order to handle imprecision and uncertainties. Pedrycz et al. [18] investigate a fuzzy set approach to estimate software project efforts. Idri et al. [12] studied the application of fuzzy logic to the cost drivers of intermediate COCOMO model. An approach using fuzzy logic was proposed by Idri et al. [11] to handle projects attributes described by categorical variables instead of numerical variables.

In a previous work, we investigate the use of crisp decision trees for software cost estimation [8, 9]. In this work, we are concerned with fuzzy decision trees models that allow exploiting the tolerance for imprecision, uncertainty and approximate reasoning offered by the fuzzy logic theory. In the present paper, we are concerned with studying the fuzzy C5, Fuzzy ID3, and FID models for software effort prediction and the impact of the pruning confidence factor and the fuzziness control threshold on the accuracy of fuzzy-based model estimates.

This paper is organized as follows: In Section II, we briefly describe the decision tree for software effort prediction. In Section III, we present the description of dataset used to perform our empirical studies. Section IV focuses on the design of the experiments. Section V present the evaluation criteria adopted to measure the predictive accuracy of the three fuzzy-based models. In section VI, we provide the results of the evaluation of the fuzzy C5, fuzzy ID3, and FID models used to estimate software development effort.

2 Decision Tree for Software Effort Prediction

Decision tree algorithm builds decision trees from a set of training data based on information gain heuristic and entropy measures to decide on the importance of the features. The main steps of the induction process of the decision tree are to calculate the entropy of each attribute to split the training set using information gain, Towing criteria, or Gain ratio and generate rules until all attribute are used or all training examples are classified; Once the tree is induced, prune it using a defined confidence limit to estimate the real error. Figure 1 shows a general approach for decision tree induction process.

International Journal of Scientific & Engineering Research, Volume [illegible]
ISSN 2229-5518

The decision tree can be interpreted by rules, each path of the branches from root to leaf can be converted into a rule with condition part represents the attributes on the passing branches from root to the leaf and the conclusion part represents the class at the leaf of the form: IF (condition 1 and condition 2 … and condition n) THEN C, where the conditions are extracted from the nodes and C is the leaf.

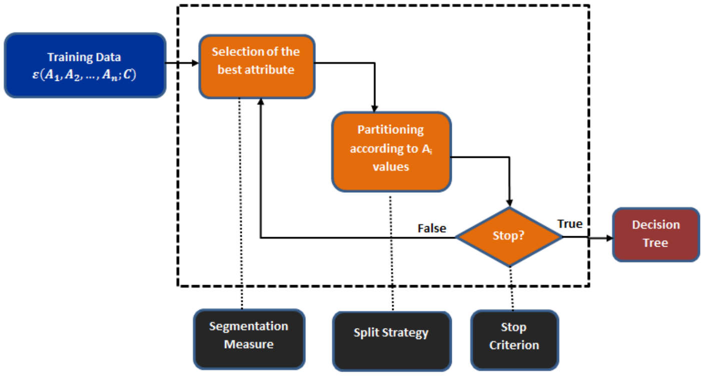

Fig.1. Decision tree induction process

Decision tree for software effort prediction is formed of one root node, which is the starting point, and a series of other nodes. Terminal nodes are leaves that represent the software effort. Each node corresponds to a split on the values of one input variable, which represents a cost driver. This variable is chosen in order to reach a maximum of homogeneity amongst the examples that belong to the node, relatively to the output variable.

3 Data Sets

Three datasets are used in this work to evaluate the prediction of our models. First dataset is The ISBSG release 8 (International Software Benchmarking Standards Group) data repository. The second dataset is Cocomo'81. The third dataset is the Web data set.

3.1 ISBSG dataset

The ISBSG repository (release 8) consists of 2027 projects collected from twenty countries around the world. The reduction of dimensionality in ISBSG dataset is primordial to operate faster and improve classification accuracy [25]. Data preprocessing operations have been performed on the ISBSG reducing the dataset dimensionality and enabling our models to operate faster and without difficulty. The preprocessing carried out on the ISBSG dataset is described on [4].

TABLE I

ISBSG Effort Factors

Metric	Definition
Function Points	Adjusted function point count number
Max Team Size	Maximum number of people on the project
Business Units	Number of business units that the system services
Locations	Number of physical locations being serviced by the system
Concurrent Users	Number of users using the system concurrently
Development Platform	Primary platform (PC. Mid-Range or Mainframe)
Normalized Work Effort	Total effort in hours recorded against the project for all phases of the development life cycle

3.2 Web dataset

The web dataset (Tukutuku) contains 53 web projects [7]. Each web application is described using nine numerical attributes such as: the number of html or shtml files used, the number of media files and team experience (see Table II).

TABLE II

Web Dataset Effort Factors

Metric	Definition
TeamExp	Average team experience with the development language(s) employed
DevTeam	Size of development team
TotWP	Total number of web pages
TextPages	Number text pages typed (~600 words)
TotImg	Total number of images
Anim	Number of animations
AV	Number of audio/video files
TotHigh	Total Number of high effort features/functions
TotNHigh	Total Number of low effort

3.3 Cocomo'81 dataset

The COCOMO'81 dataset contains 252 software projects which are mostly scientific applications developed by Fortran. Each software project is described using 13 attributes: software size measured in KDSI (Thousands of Delivered Source Instructions) and the remaining 12 numerical attributes described in Table III.

TABLE III

Cocomo'81 Effort Factors

Metric	Definition
DATA	Data base size
VIRT	Virtual Machine Volatility
TIME	Execution Time Constraint
STOR	Main Storage Constraint
TURN	Computer Turnaround Time
ACAP	Analyst Capability
AEXP	Applications Experience
PCAP	Programmer Capability

International Journal of Scientific & Engineering Research, Volume ŊǜĿÚÚÜÌĿŊǜ2Ì×ÛÌÔÉÌÙıĺŸħĦ
ISSN 2229-5518

VEXP	Virtual Machine Experience
LEXP	Programming Language Experience
SCED	Required Development Schedule

4 Design of Experiments

This section describes the experiment design of the three fuzzy decision tree based models. Fuzzy C5, FID, and Fuzzy ID3 algorithms were applied for the induction of the decision trees on three datasets: the ISBSG project data, the Cocomo'81, and on the Web dataset. The Fuzzy decision tree induction process consists on the decomposition of selected attributes into fuzzy sets, building of the fuzzy decision tree from the dataset and measure of the estimates accuracy generated by different models using the actual effort and the estimated effort.

Figure 2 illustrates the fuzzy-based models process for software effort prediction. This process consists of four phases; data preprocessing, fuzzification of the cost drivers, fuzzy induction to build the fuzzy decision tree, and model validation by the measure of the accuracy of the estimates generated by the fuzzy-based models. Each project is described by a set of attributes; the fuzzy partitions were automatically created for each attribute. The fuzzification of the software cost drivers converts crisp cost drivers into membership degrees to the different fuzzy sets of the partition. The triangular membership functions are used to represent the fuzzy sets because of its simplicity, easy comprehension, and computational efficiency [7].

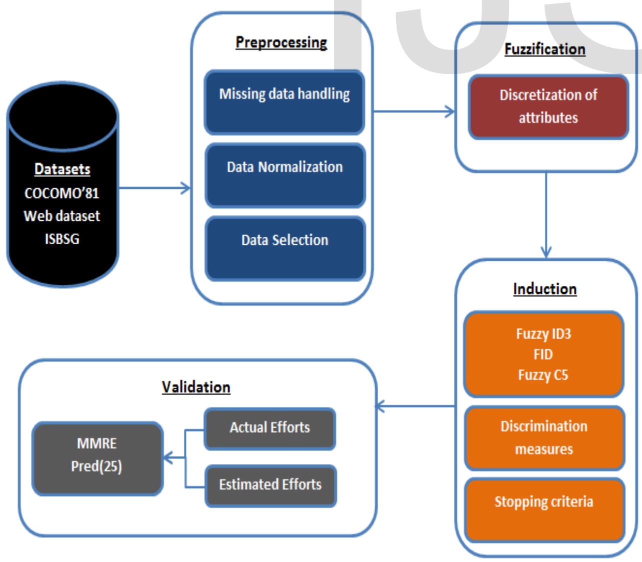

Fig. 2. Architecture of the Fuzzy Decision Tree based Models

In order to improve the model accuracy, the learning of the fuzzy decision tree must be stopped early. Our Fuzzy-based models growing phase continues until a stopping criterion is activated. Two conditions were used; the first is the fuzziness control threshold that check if the proportion of a data set of a class is greater than or equal to a threshold then it stops expanding the tree. The second is leaf decision threshold that evaluate if the number of a data set is less than a threshold then it stops expanding the tree. The value of these thresholds has great influences on the accuracy of the models. We define them in different levels in our experiments to find optimal thresholds.

4.1 Fuzzy C5 Model

Fuzzy C5 model is based on a fuzzy implementation of the C5.0 algorithm. The major characteristic of fuzzy C5 is that support fuzzy thresholds; each example belongs to a node to a certain degree. With fuzzy thresholds, both branches of the tree are investigated to give a classification that change more slowly with the value of the cost driver. Fuzzy C5 decision tree algorithm builds decision trees from a set of training data based on information gain heuristic.

Build fuzzy C5 model to estimate software development effort requires fix the model parameters. The first parameter is the pruning confidence factor (CF) and the second one is the minimum cases (MC). These two parameters were investigated in a previous work [6]. Fuzzy C5 algorithm uses the minimum cases (MC) as stopping criterion that constrains the degree to which the decision tree can grow up. Throughout the induction of the decision tree, the dataset is divided on the attributes that provide the most information gain. A series of experiments is conducted and the MC value was held constant at 2 [5]. Fuzzy C5 model was evaluated with CF values ranging from 0.1 to 1 by an increment of 0.1.

4.2 Fuzzy ID3 Model

The fuzzy ID3 is based on a fuzzy implementation of the ID3 algorithm. Fuzzy ID3 based models are grown using different values for the fuzziness control threshold that permit controlling the growth of the generated fuzzy trees [3]. The fuzziness control threshold verifies if the ratio of membership of a class at tree node is higher than a given threshold. The value for the fuzziness control threshold was varied between 0.1 and 0.9 by an increment of 0.1.

4.3 FID Model

The use of FID algorithm [21] to predict a software development effort requires the determination of the following parameters: T-norms, Inference method, Fuzzy discretization, and Stop criteria. All these parameters need to be optimized. Several values were tested and the optimal ones were used.

We have to specify the t-norm operator to use to calculate the fuzzy entropy during tree building. the minimum T-norm and product T-norm are the two commonly used fuzzy conjunction operators because of their well behavior and their computational simplicity [13]. In an earlier work [4], we have conducted multiple experiments to decide which operator to use. The results show that product t-norm perform much better in terms of predictive accuracy that the minimum t-norm. The product T-norm is used to calculate the fuzzy entropy.

To determine the class of a new project, FID performs two different methods of inference: set-based and exemplar-based [8]. Each has several manners of resolving internal conflicts (Leaf containing training data from multiple classes) and external conflicts (Multiple leaf activations with different degree

International Journal of Scientific & Engineering Research, Volume [illegible]
ISSN 2229-5518

of match). The exemplar-based inference is used to predict a new software effort.

The stopping criterion used is the fuzziness control threshold that takes on continuous values between 0 and 1. To optimize this parameter for the FID model, the value for the fuzziness control threshold was varied between 0.1 and 0.9.

5 Performance Evaluation

For this study, to evaluate the performance of estimates generated by different fuzzy decision tree models, we made use of three criteria due to their widespread relevance in software cost estimation domain [14]. They include:

Magnitude of Relative Error (MRE): MRE is value error estimated for each of the projects compared to the actual.

Mean Magnitude of Relative Error (MMRE): MMRE is used as the criteria error of the mean value of the project.

Percentage Relative Error Deviation (Pred(p)): Pred(p) is used in order to estimate the accuracy of the models.

MRE is evaluated as follows:

$$MRE = \left| \frac{Effort_{actual} - Effort_{estimated}}{Effort_{actual}} \right| \quad (1)$$

where $Effort_{actual}$ is the actual effort of a project in the dataset, and $Effort_{estimated}$ is the estimated effort that was obtained using a model or a technique.

The *MRE* values are calculated for each project in the datasets, while MMRE computes the average over *N* projects.

MMRE is evaluated as follows:

$$MMRE = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{Effort_{actual,i} - Effort_{estimated,i}}{Effort_{actual,i}} \right| \times 100 \quad (2)$$

The acceptable target values for *MMRE* are $MMRE \leq 25$.

Prediction *Pred(l)* which represents the percentage of *MRE* that is less than or equal to the value p among all projects. This measure is the proportion of the projects for a given level accuracy [19]. *Pred(pl)* is evaluated as follows:

$$Pred(p) = \frac{k}{N} \quad (3)$$

Where *N* is the total number of observations and *k* is the number of observations whose *MRE* is less or equal to *p*. A common value for *p* is 25, which is used in the present study. The prediction at 25%, *Pred(25)*, represents the percentage of projects whose *MRE* is less or equal to 25%. The acceptable values for *Pred(25)* are $Pred(25) \geq 75$.

We conducted several experiments for each model, the fuzziness control threshold was varied between 0.1 and 0.9 by an increment of 0.1. Different threshold generates different fuzzy decision trees with different classification accuracy. The aim is to find the most appropriate configuration that improves the estimates. The results for the different models have been reported.

6 Overview Of The Experimental Results

6.1 Experimental Results on Cocomo'81

This section presents and discusses the results obtained when applying fuzzyID3, FID, and fuzzyC5 to the Cocomo'81 dataset. Figure 3 and 4 show the accuracy of the generated fuzzy decision trees, measured in terms of MMRE and Pred(25), on Cococmo'81 dataset.

Table IV
Cocomo'81 Results

Cocomo'81 Pred(25)			
Threshold	**Fuzzy ID3**	**FID**	**Fuzzy C5**
0.1	95.93	80.15	40.47
0.2	95.93	79.36	50
0.3	88.24	77.38	55.15
0.4	76.92	56.47	67.46
0.5	69.68	52.38	64.28
0.6	48.41	44.44	69.04
0.7	36.51	37.3	80.15
0.8	22.62	28.57	93.25
0.9	15.08	20.63	100
Cocomo'81 MMRE			
Threshold	**Fuzzy ID3**	**FID**	**Fuzzy C5**
0.1	1.98	19.56	66.3
0.2	2.84	22.81	57.9
0.3	10.11	26.34	50.1
0.4	26.16	39.67	40.7
0.5	63.31	41.55	40.1
0.6	93.08	48.32	35.78
0.7	119.89	57.47	24.34
0.8	123.25	67.64	15.56
0.9	127.32	89.52	0

Table IV shows that the accuracy of the estimates generated by the fuzzy C5 increases with the growth of the confidence factor. For the two other models Fuzzy ID3 and FID, the accuracy of their estimates decrease with the growth of the fuzziness control threshold.

Performance of the fuzzy C5 on the Cocomo'81 dataset increase as the confidence factor increased up with a peak of 100% accuracy. Performance of the fuzzy ID3 and FID on the Cocomo'81 dataset decrease as the fuzziness control threshold increased up with a peak of 96% accuracy and 80% respectively.

International Journal of Scientific & Engineering Research, Volume ǸÔ(ÚÚÜÌ)ǸÔ2Ì×ÛÌÔÉÌÙlƳbW
ISSN 2229-5518

We have compared the average performance of the three models. The comparative study of Fuzzy ID3, FID, and Fuzzy C5 shows that the performance of Fuzzy C5 is better than Fuzzy ID3 and FID. The accuracy results of all three models are shown and according to the results the best accuracy is achieved by Fuzzy C5 with an average pred(25) of 68% and average MMRE of 36%. And then Fuzzy ID3 in the second position with an average pred(25) of 61% and average MMRE of 63%. FID remains at the last position with an average pred(25) of 53% and average MMRE of 46%.

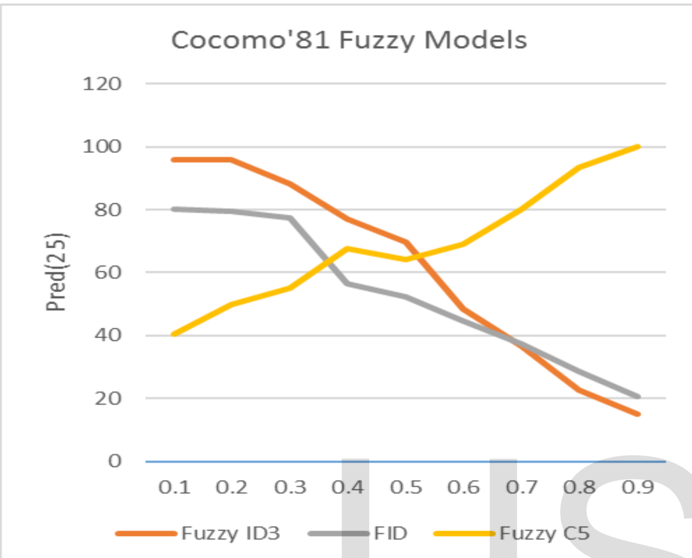

Fig. 3. Accuracy of Fuzzy ID3, FID, and Fuzzy ID3, in term of Pred(25) on Cocomo'81 dataset

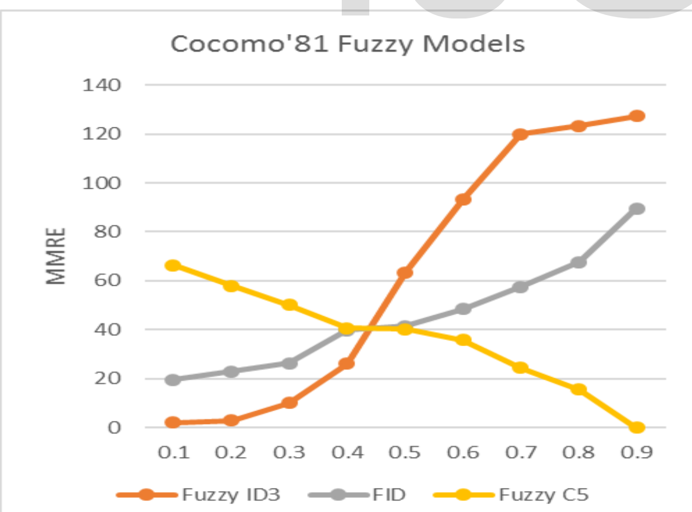

Fig. 4. Accuracy of Fuzzy ID3, FID, and Fuzzy ID3, in term of MMRE on Cocomo'81 dataset

6.2 Experimental Results on ISBSG

This section presents and discusses the results obtained when applying fuzzyID3, FID, and fuzzyC5 to the ISBSG dataset. Figure 5 and 6 show the accuracy of the generated fuzzy decision trees, measured in terms of MMRE and Pred(25), on ISBSG dataset.

TABLE V
ISBSG RESULTS

ISBSG Pred(25)			
Threshold	Fuzzy ID3	FID	Fuzzy C5
0.1	97.92	84.15	83.12
0.2	95.85	83.16	94.8
0.3	84.23	77.92	94.8
0.4	68.87	68.83	97.4
0.5	58.50	53.24	98.7
0.6	46.05	45.45	98.7
0.7	35.26	23.37	100
0.8	35.26	18.18	100
0.9	35.26	15.58	100
ISBSG MMRE			
Threshold	Fuzzy ID3	FID	Fuzzy C5
0.1	8.83	15.41	15
0.2	20.26	21.31	5
0.3	35.06	23.21	5
0.4	46.08	28.57	1.56
0.5	56.30	39.47	0.45
0.6	86.59	48.37	0.45
0.7	93.81	64.8	0
0.8	93.81	74.15	0
0.9	93.81	64.82	0

Figure 5 shows the results of the three models, in terms of Pred(25), when varying the threshold value. From this figure, we note that the accuracy of fuzzy ID3 and FID performs much better when decreasing the threshold value. For example, when setting the threshold at 0.2 the number of predictions within 25% of the actuals for Fuzzy ID3 and FID is equal to 95% and 83% respectively, and when setting the threshold at 0.6 the number of predictions within 25% of the actuals for Fuzzy ID3 and FID is equal to 46% and 45% respectively.

Figure 6 compares the accuracy of the three models, in terms of MMRE, when varying the threshold value. When setting the threshold at 0.3 Fuzzy ID3 model produces inacceptable prediction error (35%) while FID and Fuzzy C5 generate, respectively, an acceptable prediction error 23% and 5%.

It can be noticed that the accuracy of the Fuzzy ID3 is better when the threshold value is 0.3 or less (Pred(25)=84.23). When the threshold value is higher than 0.4, the MMRE and pred(25) become not acceptable. Regarding the FID model, the accuracy is better when the threshold value is less than 0.4 (Pred(25)=77.92). When the threshold is higher than 0.3, the MMRE and pred(25) become not acceptable. While Fuzzy C5 produces acceptable results for all values of the threshold.

International Journal of Scientific & Engineering Research, Volume [illegible]
ISSN 2229-5518

The comparative study of Fuzzy ID3, FID, and Fuzzy C5 shows that the performance of Fuzzy C5 is better than Fuzzy ID3 and FID. The best accuracy is achieved by Fuzzy C5 with an average pred(25) of 96% and average MMRE of 3%. And then Fuzzy ID3 in the second position with an average pred(25) of 61% and average MMRE of 59%. FID remains at the last position with an average pred(25) of 52% and average MMRE of 42%.

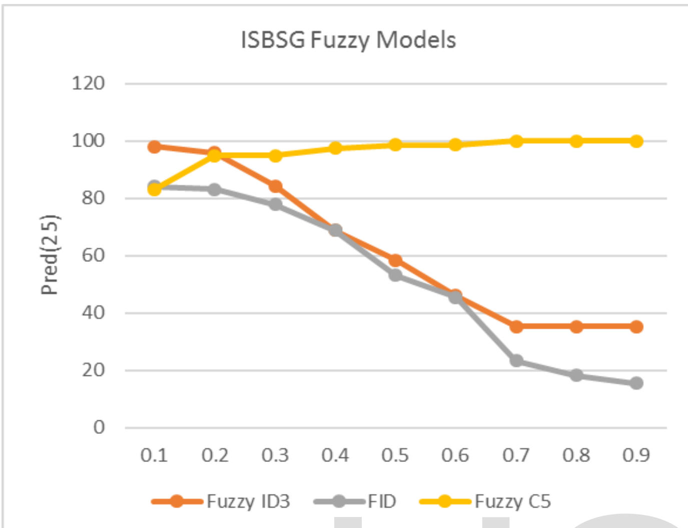

Fig. 5. Accuracy of Fuzzy ID3, FID, and Fuzzy ID3, in term of Pred(25) on ISBSG dataset

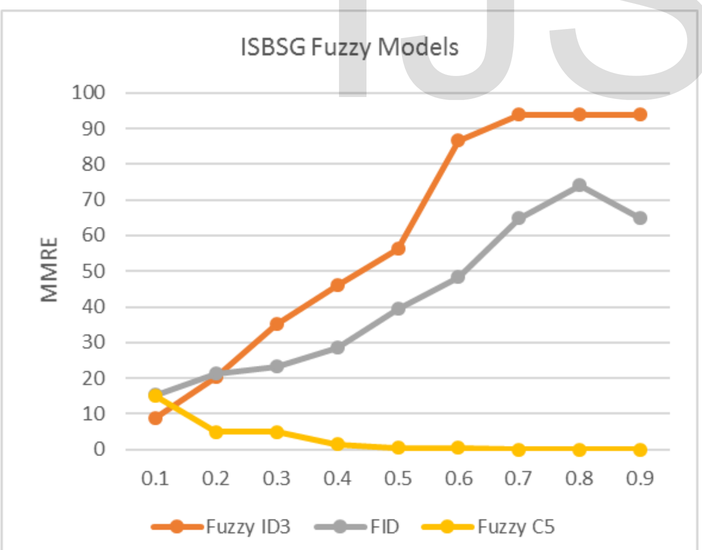

Fig. 6. Accuracy of Fuzzy ID3, FID, and Fuzzy ID3, in term of MMRE on ISBSG dataset

6.3 Experimental Results on Web Dataset

This section presents and discusses the results obtained when applying fuzzyID3, FID, and fuzzyC5 to the Web dataset. Figure 7 and 8 show the accuracy of the generated fuzzy decision trees, measured in terms of MMRE and Pred(25), on Web dataset.

TABLE VI
WEB DATASET RESULTS

Web Dataset Pred(25)			
Threshold	Fuzzy ID3	FID	Fuzzy C5
0.1	97.73	98.11	88.68
0.2	93.18	94.34	94.34
0.3	95.45	90.57	94.34
0.4	90.91	75.47	94.34
0.5	90.91	64.15	94.34
0.6	83.02	56.6	94.34
0.7	50.94	45.28	100
0.8	45.28	41.57	100
0.9	20.75	34.89	100
Web Dataset MMRE			
Threshold	Fuzzy ID3	FID	Fuzzy C5
0.1	5.31	2.38	21.95
0.2	1.82	4.3	6.78
0.3	3.87	7.09	6.78
0.4	5.82	19.27	6.78
0.5	9.09	49.3	6.78
0.6	90	52.19	0
0.7	97.41	51.68	0
0.8	111.83	59.23	0
0.9	176.99	65.67	0

Figure 7 shows the results of the three models, in terms of Pred(25), when varying the threshold value. From this figure, we note that the accuracy of fuzzy ID3 and FID performs much better when decreasing the threshold value. For example, when setting the threshold at 0.1 the number of predictions within 25% of the actuals for Fuzzy ID3, FID, and Fuzzy C5 is, respectively, equal to 97%, 98%, and 88%. And when setting the threshold at 0.6 the number of predictions within 25% of the actuals for Fuzzy ID3, FID, and Fuzzy C5 is equal to 83%, 56%, and 94%, respectively.

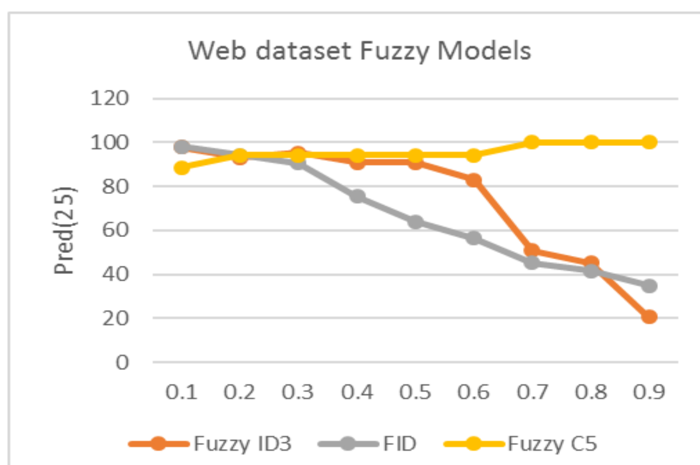

Fig. 7. Accuracy of Fuzzy ID3, FID, and Fuzzy ID3, in term of Pred(25) on Web dataset

International Journal of Scientific & Engineering Research, Volume ÑÒ(ÚÚÜÌ ÑÒ2Ì×ÛÌÔÉÌÜıl͸
ISSN 2229-5518

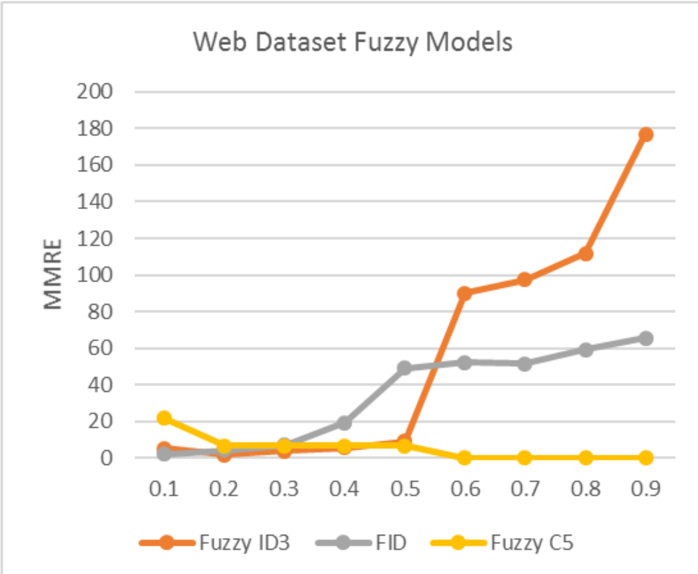

Fig. 8. Accuracy of Fuzzy ID3, FID, and Fuzzy ID3, in term of MMRE on Web dataset

Figure 8 compares the accuracy of the three models, in terms of MMRE, when varying the threshold value. When setting the threshold at 0.5 FID model produces inacceptable prediction error (49%) while Fuzzy ID3 and Fuzzy C5 generate, respectively, an acceptable prediction error 9% and 6%. Fuzzy C5 achieves the best accuracy with an average MMRE of 5%. Then FID in the second position with an average MMRE of 34%. Fuzzy ID3 remains at the last position with an average MMRE of 55%.

6.4 Best Results

This section represents the best accuracy produced by the three investigated models. Fuzzy ID3, FID, and Fuzzy C5 were compared using two main measures: the size of the tree and the accuracy of the model. Total number of nodes in the tree measures tree size. The accuracy of the model is measured by using the MMRE and the Pred(25).

The fuzzy C5 model performs much better when increasing the confidence factor value while the fuzzy ID3 and FID models perform much better when decreasing the fuzziness control threshold value. This divergence can be explained by the facts that the fuzziness control threshold growth may lead to a small tree that explains the decrease of the classification accuracy. A lower confidence factor value reduces the generated tree while the classification accuracy will be lower.

The results obtained using the fuzzy C5 show that lowering the pruning confidence factor is useful for reducing the tree size, and helps in filtering out inappropriate nodes that would otherwise lead to classification errors. While, the results obtained using the fuzzy ID3 and FID show that lowering the fuzziness control threshold lead to a large tree and to an overfitting.

Table VII shows the best accuracy of the estimates generated by the fuzzy ID3, FID, and Fuzzy C5. On Cocomo'81, Fuzzy C5 model generates the smallest tree (48 nodes) with the highest accuracy (88%). FID generates a moderate tree of 56 nodes with an acceptable accuracy of 77%. Fuzzy ID3 produces a large tree made up of 64 nodes with an accuracy of 76%.

TABLE VII
ACCURACY OF BEST MODELS

Cocomo'81			
	FuzzyID3	FID	FuzzyC5
MMRE	26.16	26.34	24.34
Pred(25)	76.92	77.38	80.15
Size	64	56	48
Web			
	FuzzyID3	FID	FuzzyC5
MMRE	90	19.27	21.95
Pred(25)	83.02	75.47	88.68
Size	33	35	23
ISBSG			
	FuzzyID3	FID	FuzzyC5
MMRE	44.36	23.21	15
Pred(25)	80.08	77.92	83.12
Size	78	93	78

On the web dataset, the largest tree is generated by FID model (35 nodes) and it has the lowest accuracy (75%) compared to Fuzzy ID3 and fuzzy C5. Fuzzy ID3 model produces a similar tree size (33 nodes) with higher accuracy (83%). The smallest tree is generated by Fuzzy C5 (23 nodes) with a significant accuracy (88%).

On the ISBSG repository, Fuzzy ID3 and Fuzzy C5 generate the smallest trees (78 nodes) compared with FID (93 nodes). FID model produces the lowest accuracy (77%), while Fuzzy C5 produces the highest accuracy (83%).

7 CONCLUSION

Fuzzy based models are investigated using Cocomo'81, Web dataset, and ISBSG repository in this paper. A comparative study was conducted with different fuzzy-based algorithms; Fuzzy ID3, FID, and Fuzzy C5. The results show that combining fuzzy logic and the decision tree models improves greatly the accuracy of estimates. In our testing, we found that proper utilization of the confidence factor and the fuzziness control threshold has shown an increase of estimation accuracy. Therefore, several values for the thresholds must be evaluated when building fuzzy decision trees for software effort prediction to find the most appropriate value for the study dataset.

International Journal of Scientific & Engineering Research, Volume ɈŇ(ÚÚÜÌɈŇ2Ì×ÛÌÔÉÌÙıl ɣhɯ
ISSN 2229-5518

REFERENCES

[1] Breiman, L. Friedman, J.H. Olsen, R.A. & Stone, C.J. Classification and Regression Trees, Wadsworth, 1984.

[2] Berger, H. Merkl, D. and Dittenbach, M. Exploiting Partial Decision Trees for Feature Subset Selection in e-Mail Categorization, in Proceedings of the 2006 ACM Symposium on Applied Computing (SAC 2006), Dijon, France, 2006, pp. 1105-1109.

[3] Elyassami, S. and Idri, A. Applying Fuzzy ID3 Decision Tree for Software Effort Estimation, International Journal of Computer Science Issues (IJCSI), Vol. 8, Issue 4, No 1, July 2011, pp. 131-138

[4] Elyassami, S. and Idri, A. Investigating effort prediction of software projects on the ISBSG dataset. International Journal of Artificial Intelligence & Applications (IJAIA), Vol.3, No.2, March 2012, pp.121-132.

[5] S. Elyassami and A. Idri, 2013. Evaluating software cost estimation models using fuzzy decision trees, Recent Advances in Knowledge Engineering and Systems Science, ISBN: 978- 1-61804-162-3, pp. 243-248.

[6] S. Elyassami and A. Idri, 2013. Fuzzy model for an early estimation of software development effort, International Journal of Applied Mathematics and Informatics, Issue 3, Vol.7, pp. 69- 77.

[7] S. Elyassami and A. Idri, 2011. A fuzzy Decision Tree to estimate development effort for Web Application, International Journal of Advanced Computer Science and Applications - Special issue on artificial Intelligence, pp. 87-91.

[8] A. Idri and S. Elyassami, 2010. Software Cost Estimation Using Decision Trees, Proceeding of Sixième Conférence sur les Systèmes Intelligents: Théories et Applications (SITA'10), pp. 120-125.

[9] A. Idri and S. Elyassami, 2010. Web Effort Estimation Using Decision Trees, Proceeding International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), pp. 224-228.

[10] Finnie, G. R. and Witting, G. and Desharnais, J.-M. "A Comparison of Software Effort Estimation Techniques: Using Function Points with Neural Networks, Case-Based Reasoning and Regression Models", Systems and Software, Vol. 39, No. 3, 1997, pp. 281-289.

[11] Idri, A. and Abran, A. "Evaluating Software Project Similarity by using Linguistic Quantifier Guided Aggregations" Proceedings of IFSA/NAFIPS 2001, Vancouver,Canada, pp. 470 - 475, 2001.

[12] Idri,A. Kjiri, L. and Abran,A. "COCOMO Cost Model Using Fuzzy Logic", 7th International Conference on Fuzzy Theory & Technology, Atlantic City, NJ, February, 2000. pp. 219-223.

[13] E. H. Mamdani, "Applications of Fuzzy Set Theory to Control Systems: A Survey," in Fuzzy Automata and Decision Processes, M. M. Gupta, G. N. Saridis and B. R. Gaines, eds., North-Holland, New York, 1977, pp. 1-13.

[14] Korte M. and Port, D. Confidence in Software Cost Estimation Results Based on MMRE and PRED, PROMISE'08, May 12-13, 2008, pp . 63-70.

[15] Molokken, K. and Jorgensen, M. A Review of Surveys on Software Effort Estimation, in International Symposium on Empirical Software Engineering, 2003, pp. 223-231.

[16] Nisar, MW. and Wang, Y.-J. and Elahi, M. "Software Development Effort Estimation Using Fuzzy Logic – A Survey", in 5th International Conference on Fuzzy Systems and Knowledge Discovery, 2008, pp. 421-427.

[17] Pedrycz, W and Sosnowski, Z. A., "The design of decision trees in the framework of granular data and their application to software quality models," Fuzzy Sets and Systems, Vol. 1234, 2001, pp. 271-290.

[18] Pedrycz, W, Peters, J.F and Ramanna, S, A Fuzzy Set Approach to Cost Estimation of Software Projects, Proceedings of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering Shaw Conference Center , Edmonton Alberta, Canada. 1999, pp. 1068-1073.

[19] Quinlan, JR. "Induction on decision tree," Machine Learning, Vol. 1, 1986, pp. 81-106.

[20] Quinlan, JR. C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[21] Rulequest Research, Data Mining Tools See5 and C5.0, http://www.rulequest.com/see5-info.html.

[22] Sharma, V. and Verma, H. K. "Optimized Fuzzy Logic Based Framework for Effort Estimation in Software Development", Computer Science Issues, Vol. 7, Issue 2, No. 2, 2010, pp. 30-38.

[23] Shepperd M. and Schofield. C. "Estimating Software Project Effort Using Analogies." Transactions on Software Engineering, vol. 23, no. 12, 1997, pp. 736-747.

[24] Selby, R. W. and Porter, A.A. "Learning from examples: generation and evaluation of decision trees for software resource analysis", IEEE Transactions on Software Engineering, Vol. 14, No. 12, 1988, pp. 1743-1757.

[25] Twala, B. Cartwright M. and Shepperd, M. "Comparison of various methods for handling incomplete data in software engineering databases", Proc. Intl Symp Empirical Soft Eng 2005, 105-114.