International Journal on Future Revolution in Computer Science & Communication Engineering ISSN: 2454-4248
Volume: 4 Issue: 3 429 – 440

Hybrid Parameter Optimization Approach with Adaptive Neuro Fuzzy Inference System for the Software Maintainability

P.R Therasa
Computer Center
A.C. Tech, Anna University
Chennai, India
therasa.peter@gmail.com

P.Vivekanandan
Dept. of Chemical Engineering
A.C. Tech, Anna University
Chennai, India
pvivek46@yahoo.com

***Abstract*—**This paper presents a novel method to measure the maintainability of the software from the design artifact. It is an inevitable measure because it aims to attain software with a better quality. The system is designed to measure the maintainability of the system from the UML class metric. This is extracted from the UML class diagram to predict the maintainability of the class diagram. The system is implemented using CFS from the Weka tool to select an optimized variable from a set of variables i.e UML class metric. Hybrid ANFIS is an artificial intelligence technique which has been incorporated with the optimizing algorithms to reduce the overall number of UML metric and build a Fuzzy Inference System (FIS) based on the learning process. The optimization attains an enhanced result since it is done continually by both using feature selection and optimization algorithms repetitively, which results in reducing the UML metric considerably to measure the maintainability of the software. The proposed research work is evaluated in terms of the performance measures, MSE, RMSE, true positive rates and the result is clearly shown that a better optimization of the maintainability measure estimation process can be done.
Keywords:*ANFIS,Feature selection, Fuzzy Logic, Software Maintainability,UML.*

*****

I. INTRODUCTION

The maintainability is considered as non-functional quality attribute of software and it is not only a measure at the end of the development but it also being measured in the design level. The software maintainability is easy with the software can be modified or updated in the operation stage. The software maintenance is defined as a "*modification of a software product after the delivery to correct the faults, improve the performance or other attributes"* [1].There are some of the main things need to be considered while maintaining the software such as the readability of the software, the understandability and the proper documentation of the software [27].The readability of the software is the degree to which a reader can quickly and easily understand source code. Aggarwal et al. [27] deal with the main factors that affect the software maintainability. The readability of the source codes are based on the comment ratio and line of code, documentation is based on Gunning's Fox Index. According to Laitinen's [28] understandability is measured based on the term of the language of the software, it includes all the tokens rather than the keywords or reserved words.

Modifying the software in operation stage is the time consuming and very costly comparing in other phases of the software development. The maintainability is measured based on the internal quality attribute [15],[26] of the software. So, there is a need to predict the maintainability of the software in design stage and redesign the system if it is possible. To analyse the design, the design document is very important. The documentation benefits for updating and revising the software and it has been commonly studied [4-5]. There are approximately 20 percent of the maintenance task time will be reduced in the proper documentation [23].UML [24] diagram acts as the formal modeling representation used as the supplementary documentation in a pictorial scheme. In industrial areas, UML is one of the most commonly used modeling technologies.

There are many advantages of the UML diagrams. The UML diagrams are useful in software development and maintenance with a high level of detail and low level of detail [10] respectively. It reduces the defect density as it improves the functional correctness [32].It improves the traceability of the software from the requirements to the code [1].Despite of its merits, there are some of the disadvantages of using the UML diagram such as the sufficient training in UML is needed to use the forward-looking features, language rules and semantics of the attributes [1].The time used up for updating the UML diagrams conferring to the alterations in source code respond the development in source code maintenance time [12],[17].The UML models created in the requirements analysis process the influence neither the comprehensibility of source code nor its modifiability [25].There are some of the evidences to UML as it reduces the maintainability cost[22]. Here, the UML is taken as the design artifact to measure the maintainability of the software.

By using a hybrid learning algorithm that combines gradient method with the least squares estimate for parameter identification, the ANFIS can perform input-output mapping based on both human knowledge (in the form of fuzzy if-then rules) and the stipulated input-output data pairs [3]. However, despite the successful use of ANFIS to solve these nonlinear problems, a left behind issue is identifying the most suitable membership functions while simultaneously optimizing the premise and consequent parameters. Therefore, the proposed method with the use of CFS [30] to address the above limitations of the conventional ANFIS and developed a CFS-based ANFIS approach for solving the parameter optimization problems [29]. As existing research states that a number of models has been developed with the various parameters to measure the maintainability. This research uses two main objective of optimizing the design parameters [32] in terms of better maintainability value and train the system with selected parameter in ANFIS. The previous study was based on trial-and-error design method and the Taguchi-based design method [18].Thus, in this study, the CFS based ANFIS is proposed to design the parameters for a maintainability measurement. The resolution proposed in this paper is CFS-based ANFIS to measure the maintainability.

To optimize the parameters for the maintainability from the UML class diagram, the proposed approach combines data collection from Genero [12], optimization methods, designing the model with selected variables and training with ANFIS. Two main parts of this model are parameter optimizing and designing the system with the best parameter. After setting the design parameters and output, ANFIS is used for carrying out the training and creating of new FIS and a new membership function. Finally, the hybrid learning algorithm is used to optimize the parameters in the premise and consequent parts. Thus, CFS is used to optimize the design parameters in terms of easy of the maintainability. The coupled methodology to optimize the design parameters is then tested with a testing data set and the results are discussed. The proposed systematic method is indeed to obtain superior design parameters compared to approaches recently reported in the literature.

The maintainability predication measure identifies how much effort needs to maintain the software using UML design artifact. The UML metric, hence, has an important role to play in the identification of the maintainability of the software of an application. An algorithm has been proposed that prioritizes the UML metric based on the features that impact the maintainability of the system.

The organization of the paper is structured as follows. Section I defines the introduction about the maintainability measure from UML with ANFIS and CFS. Related work is presented in the section II. The proposed approaches and preliminary work are introduced in Section III. Section IV presents the system description. Section V shows the experimental set up and section VI explains the results analysis. Finally, Section VII concludes the study..

II. Related Work

The Software Maintainability based on the UML metric Genero et al. [12] conducted a controlled experiment which has 11 class diagram UML metric for size and complexity to measure the maintainability. Yi and Wu [36-37] evaluated the maintainability from the sub-characteristic understandability and modifiability time for the UML class diagram. The UML class diagram is given to the university students who know the software engineering concepts. Based on the controlled experiment results and fuzzy regression analysis of the UML class metric, they have found that aggregation and generalization relations are highly correlated with the understandability, the modifiability and the analyzability time.

In [20], the structural complexity [24] of the class diagram is measured based on the entropy by converting the class diagram into the weighted class dependency graph method.

Zhou and Xu [23-24] analyzed the relationships between design metrics and the maintainability of the software. They also bring into the existence that the size and complexity metrics are strongly correlated to the software maintainability, compared to coupling and cohesion. Alshayeb [25] examined the relationship between the stability metrics and indices of the maintenance effort by the empirical study. They have concluded that the classes with the higher values of Class Stability Metrics (CSM) are related to lower values of the maintenance effort. The effort in hours is the measurement of the maintenance. Only the CSM is correlated with the maintainability measured by the number of altered lines. In [5-6], a machine learning approach to measure the maintainability of the software is performed.

A. *CFS with parameter optimized*

Using CFS the parameters are optimized by applying the correlation coefficient values on all the parameters. The relevant parameter which is matched with the output data is selected as the optimized parameter.

In [29], they presented the comparison techniques between the attribute selection based on CFS and consistency on the lookout for better method which possibly with the ANFIS. Best model will be used for forecasting business bankruptcy in Thai enterprises. They are affording two models for their study, one predicting model is CFS-ANFIS and another one was based consistency ANFIS. The result shows that error rates obtained from CFS-ANFIS are lower than the error rates obtained from consistency ANFIS.

III. Preliminaries

A. *Parameter selection using CFS*

The machine learning algorithm learns from the benchmarking results, against the best data available in the other research challenges. It gives as a stack of knowledge into how one can perform against the best on a level in concert field. At first, one used to trust that the machine learning will be about an algorithm. One should to know which one to apply to go ahead the best. It was not the situations to the conquerors were utilizing similar algorithm in which a considerable measure of other individuals were utilizing. There are two things which recognize best from others in a large portion of the data are Feature Creation and Feature Selection. As such, it comes down to making factors which catch and resolve the correct decisions about which variable to decide for the proposed models. Both these aptitudes require a huge amount of training of the preferred parameters.

*1) Feature Selection:*The Machine learning algorithm is working based on the rules. This turns out to be considerably more necessary when the quantities of parameters are huge. It requires not to utilize each available element but to use the best parameter for execute the algorithm by strengthening in just those elements that are truly impact. The subsets give the preferred outcomes over the total number of features for a similar method. It lessens the preparation time and the assessment time of experiment. Thus, the feature selection gives the following: It permits the machine learning to train quicker, reduces the complexity and choses the correct subset to make the model perfect and declines the overfitting.

2) Filter Method

The feature selections are based on the scores obtained from the statistical methods with the outcome. It commonly used as a preprocessing of feature to obtain the best features. The correlation is a main general term here. The following diagram shows the steps involved in the feature selection.

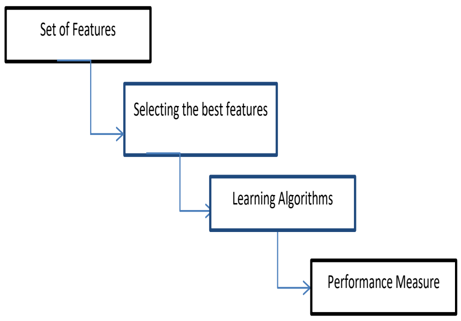

Figure1. Feature Selection

Pearson's Correlation: It is used to quantify the linear dependence between two continuous variables X and Y. This value varies from -1 to +1. Pearson's correlation is given as:

$$\rho_{xy} = \frac{cov\,(xy)}{\sigma_x \sigma_y} \qquad (1)$$

LDA: Linear discriminant analysis is used to characterize or separate two or more classes (or levels) of a categorical variable.

ANOVA: ANOVA stands for Analysis of variance. It has more than one categorical independent features and one continuous dependent feature. It gives a statistical test response means of groups are equal or not.

Chi-Square: This statistical test applied to test the likelihood of correlation using the frequency distribution.

3) *Wrapper Methods*

In this methods, a subset of features are used to train the model. According to the extrapolations that achieved from the former one, decision has to be taken whether to include or eliminate the features. It is necessarily essential to reduce a search problem. Some of the common examples of wrapper methods are forward feature selection, backward feature elimination, recursive feature elimination, etc.

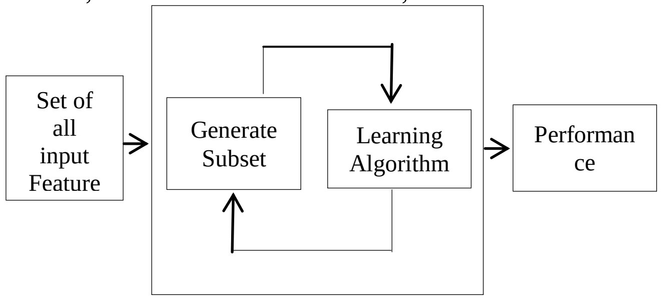

Figure2. WrapperMethod

B. *Attribute Selection and Evaluation in Weka*

There are two main processes in the Weka tool to select the optimized parameters which are attribute evaluator and search methods [31]. Using the attribute evaluator, the subsets are assessed and using search method, the space of possible subsets is searched.

The second approach, a correlation-based feature selection (CFS) method, is a state-of-the-art algorithm implemented through the CfsSubsetEval algorithm in the WEKA software package by Hall et al.[30]. It is a fully automatic algorithm that does not require predefined thresholds or the number of features. The algorithm ranks the parameter based on the subset evaluating method. According to that a correlation-based heuristic evaluation function has been executed, it retains relevant parameter that are strongly correlated with the output class. Irrelevant data with the small correlations are screened.

1) *Attribute Evaluator*

The Attribute Evaluator is a method by which the subsets of attributes are assessed. It is a way of assessing the structure of the model and finding the perfection of the model. Some of the attribute evaluation methods are listed out below:

CfsSubsetEval: It selects the highly correlated with the maintainability class value.

ClassifierSubsetEval: It uses the predictive algorithm to select the parameter.

WrapperSubsetEval: It combines' a classifier and n fold validation to obtain the parameter.

2) *Search Method*

The Search Method is the standardapproaches in which the subsets are navigateon the search space based on the subset evaluation. Two baseline methods are Random Search and Exhaustive Search. Even though thegraphs search algorithms are popular such as Best First Search. Some of the examples of the attribute evaluation methods are:

Exhaustive: It uses all combinations of attributes.

BestFirst: A best-first search strategy is used.

GreedyStepWise: A forward (additive) or backward (subtractive) strategy is used.

C. *Adaptive Neuro Fuzzy Inference System (ANFIS)*

ANFIS is a functional technique which uses for integration of the Artificial Neuron Network and Fuzzy Logic together. With this technique, ANFIS can reimburse the limitation of one procedure with an advantage of another one. In addition, the ability to learn and adapt from the data characteristic is the advantage of Artificial Neuron Network, but this procedure has a critical constraint about the explanation of in-depth learning which every computer programs cannot understand the learning dimensions as good as human can. According to ANFIS technique, this limitation can be compensated by the fundamental stage of Fuzzy Logic which is developed from the if-then rule in the Crisp Logic decision to be the Fuzzy decision [2-3]. The ANFIS has many layer components in it [2,3]. And the figure of layers components is given as follows: The ANFIS has six layer parts in it. The further description of the figure of layers logical parts are shown in the figure 3.

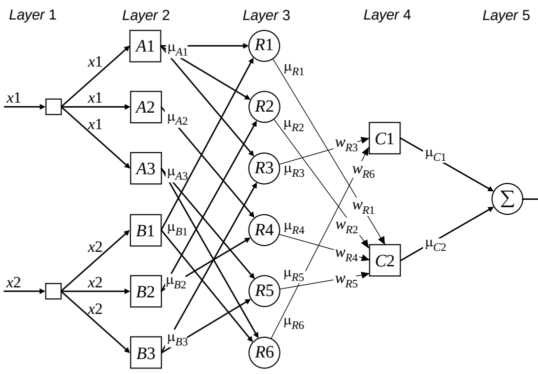

Figure3. ANFIS 6 layer Architecture

TABLE I ANFIS LAYERS

Layers	Description
Layer1	The first layer is the Input layer, replaces the data input with x which has a number of dimensions. All the input(neuron) in this layer will sent the external crisp input signals directly to the next layer. That is, $y_i^{(1)} = x_i^{(1)}$ (2)
Layer2	Second layer is the Fuzzification layer, this layer is used for Fuzzy valuation under different membership functions. A triangular membership function can be specified by two parameters $\{a, b\}$ as follows: $y_i^{(2)} = \begin{cases} 0, & \text{if } x_i^{(2)} \le a - \frac{b}{2} \\ 1 - \frac{2\lvert x_i^{(2)} - a\rvert}{b}, & \text{if } a - \frac{b}{2} < x_i^{(2)} < a + \frac{b}{2} \\ 0, & \text{if } x_i^{(2)} \ge a + \frac{b}{2} \end{cases}$
Layer3	The third layer is the layer of Fuzzy rule. This layer integrates all the output from the second layer with specified Fuzzy rules from the given equation below. For instance, neuron *R*1, which corresponds to *Rule* 1, receives inputs from neurons *A*1 and *B*1.In a neuro-fuzzy system, intersection can be implemented by the product operator. The output of the third layer is in the eq.4 and eq.5. $y_i^{(3)} = x_{1i}^{(3)} \times x_{2i}^{(3)} \times \ldots \times x_{ki}^{(3)}$ (4) $y_{R1}^{(3)} = \mu_{A1} \times \mu_{B1} = \mu_{R1}$ (5)
Layer4	The Fourth layer is the Normalization layer. In this layer the output membership neuron combines all its inputs by using the fuzzy operation **union**. $y_i^{(4)} = x_{1i}^{(4)} \oplus x_{2i}^{(4)} \oplus \ldots \oplus x_{li}^{(4)}$ (6) $y_{C1}^{(4)} = \mu_{R3} \oplus \mu_{R6} = \mu_{C1}$ (7)
Layer5	This layer is called the Defuzzification layer. In this layer the output fuzzy set is given, respective rules forms integrated firing strengths and combine them into a single fuzzy set. Thus, the formula of the weighted average of the centroids of the clipped fuzzy sets *C*1 and *C*2 is calculated as $y = \frac{\mu_{C1} \times a_{C1} \times b_{C1} + \mu_{C2} \times a_{C2} \times b_{C2}}{\mu_{C1} \times b_{C1} + \mu_{C2} \times b_{C2}}$ $y_i^{(5)} = x_i^{(5)} \left[k_{i0} + k_{i1}\, x1 + k_{i2}\, x2\right] = \bar{\mu}_i \left[k_{i0} + k_{i1}\, x1 + k_{i2}\, x2\right]$ $x_i^{(5)}$ and $y_i^{(5)}$ are the input and output of defuzzyfication neuron i in the layer 5 respectively. andk_{i0}, k_{i1} and k_{i2} are a set of consequent parameters of rule *i*.
Layer6	The Final layer is called the layer of Neuron Summarization or ANFIS output layer. This output is obviously appeared in the form of Fuzzy Sugeno function which can be calculated based on the input x and the set of consequent parameters k. The form of ANFIS output can be showed as: $y = \sum_{i=1}^{n} x_i^{(6)} = \sum_{i=1}^{n} \bar{\mu}_i \left[k_{i0} + k_{i1}\, x1 + k_{i2}\, x2\right]$

IV. SYSTEM DESCRIPTION

A. System Architecture

Architecture Diagram for selecting optimized parameter for the maintainability measure is given in the figure 4. The 11 UML class metric with the maintainability value from genero[8] are given as input to CFS. The CFS is used here to reduce the irrelevant feature and gives out the relevant feature set to build the best system to measure the maintainability. Then by using the relevant feature set, the ANFIS model has been developed.

The relevant feature set is given as input to the ANFIS in MAT Lab to generate the FIS with a new set of rules from the training dataset. The developed ANFIS model is further trained with the different methodologies and it is to develop the new FIS and new rules that are the trained FIS model. The trained FIS is now tested with the test dataset given as input.

The performance evaluation is done by comparing the results of the generated new SFIS and trained SFIS and also the relevant parameter gives the better result compared to the inclusion of the parameters to measure the maintainability of the software from the UML class diagram.

B. Methodology

First, some of the features are dropped from the number of features which are not relevant to reduce the overfitting and improve the generalization of the models and to gain a better understanding of the features and their relationship to the response variables. Second one is the building the relation between the selected parameter to achieve the better result. There are 11 input parameters from the class metric and one output maintainability value. From the 11 parameters, the CFS selected three set of relevant parameters.

The selected UML metric are further scaled down using the Hybrid Neuro Fuzzy Inference System (ANFIS) called “Hybrid ANFIS”. Thus, the optimal UML metric are selected for further optimization by the ANFIS. This is done by fuzzy logic principles that select only the UML metric that are needed for validating the changes in the software for the maintenance. The validation of the system is done by the RMSE, error rate and truth positive rates.

C. Data Preparation and Sample Size

This paper uses the data set from genero model [8] for estimating the maintainability of the software from UML class metric based on controlled experiment. It has about 28 class diagram, the maintainability value is estimated from the sub-characteristics [8-11]. The table II - table V show the lists of attributes which give the values of UML class metric. Sample size of this study is listed metric in the tables. As the values are taken from the controlled experiments, there is no need of normalization of the values. All the available data sets are taken for the experiments.

1) Data set Preparation

Initially, the collected data should be divided into 2 sets which are the training data and the testing data set under the proportion of 50 percent for the training and 50 percent for the testing. There are 14 UML metrics in the training dataset and 14 UML metrics from the class diagram for the testing dataset. The classified dataset is stated in the following table II-table V.

There are 11 input parameters from the class metric and one output maintainability value. From the 11 parameters, the CFS selected three set of relevant parameters.

D. Model testing and comparison

The most extensive form of validation is in the form accuracy, sensitivity, specificity, precision and recall. The average error was computed and also used to evaluate the model. The root mean square error (RMSE), Root mean square error (RMSE), mean error (ME) and coefficient of determination (R2) were calculated to determine the model performance [16].

Accurate predictions have an RMSE close to zero. When the RMSE is standardized by the mean, it is the CV-RMSE. The ME represents the bias of prediction and should be close to 0 for the unbiased methods. The R2 value measures how well the predicted values approximate the observed data values. An R2 value 1 indicates that the regression line fits the data perfectly.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(a_{obs,i} - p_{model,i})^2}{n}} \quad (11)$$

where n is the number of UML class diagrams taken for the prediction and a represent the actual value of the maintainability and p represents the predicted value.

$$\mathbf{R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}} \quad (12)$$

$$\boldsymbol{Adj\ R^2 = 1 - \frac{(n-1)}{(n-p)}\frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{n}(y_i-\overline{y})^2} = 1 - \frac{(n-1)}{(n-p)}(1-R^2)} \quad (13)$$

where n is the number of observations , y^is the predicted value of y to the comparison of the observed and the predicted values, is the mean of the observations y.

V. EXPERIMENTAL SET UP AND RESULTS DISCUSSION

A. Experimental set up

Three types of experiments were conducted with use of the Weka tool to select the best parameter. The first one was done with the five parameters, second one with the six parameters and third one with the seven parameters. The evaluation method is the CfsSubsetEval and to the search best subset the method used is the BestFirst.

B. Attribute Selection with the Five and Six Parameters

The optimized attributes are selected from the set of attributes in the UML class metric. The weka tool is used to select the parameter with evaluator CFS (correlation Feature Selection) subset evaluation method. It uses the BestFirst search method. The following parameters are selected from the given 12 parameters from the 28 class diagrams.

Parameter Selection from the Weka is given below:

Evaluator method: weka.attributeSelection.CfsSubsetEval -P 1 -E 1

Method of Search: weka.attributeSelection.BestFirst -D 1 -N 5

Data File Relation: wekatest

Total Number of Instances: 28

Total Number of Attributes: 12 (NC NA NM NAssocNAggNDepNGenNAggHNGenHMaxHaggMaxDITGeneroModel (class attribute))

The evaluation mode uses all the training data set .The Attribute Selection is based on all the input data sets. It uses Best first Search Method:

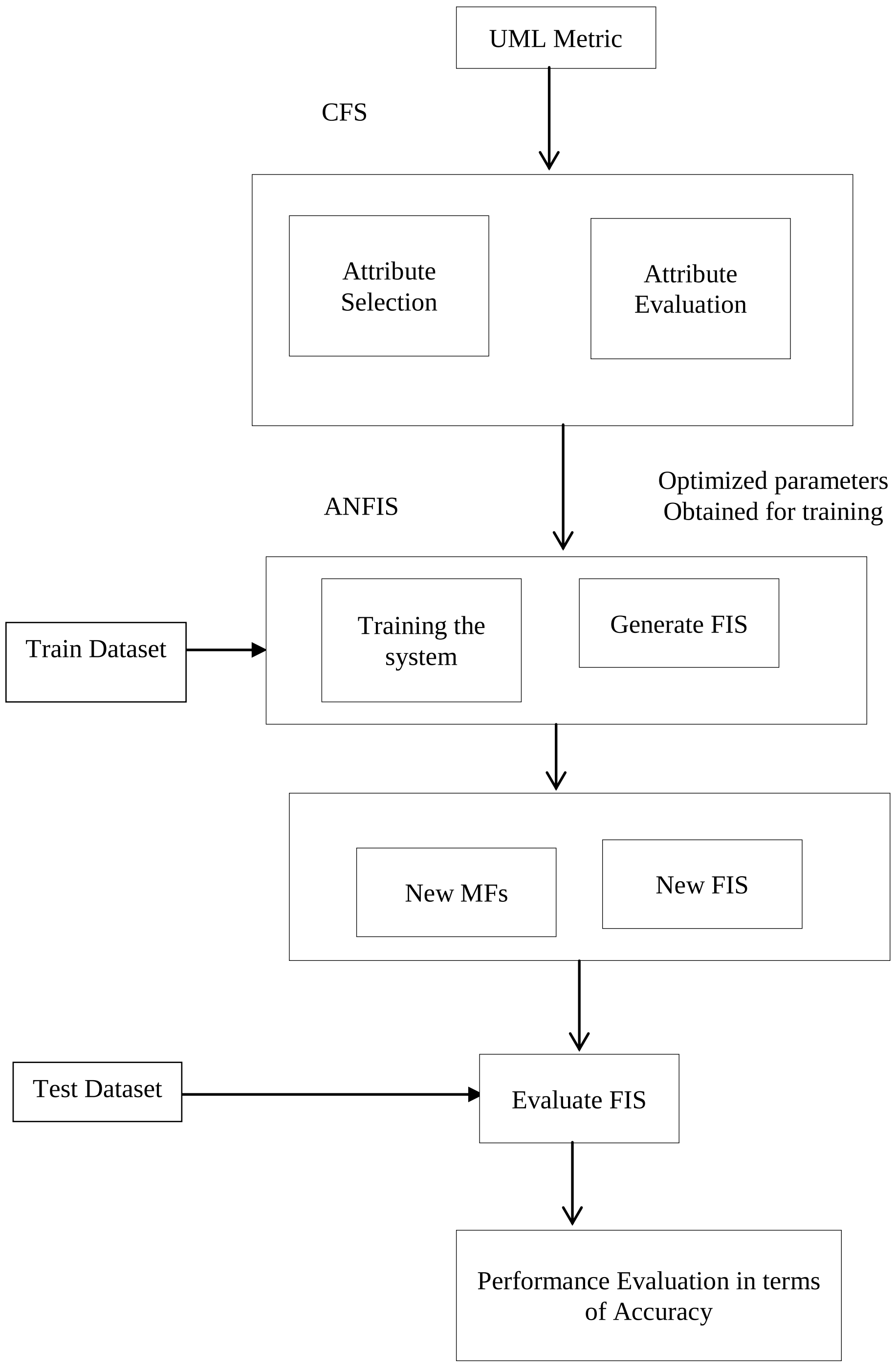

Figure4. Architecture Diagram.

Start set: No attributes
Search direction: forward
Total number of subsets evaluated: 64
Merit of the best subset found: 0.892
Attribute Subset Evaluator (supervised, Class (numeric): 12 Genero Model): CFS Subset Evaluator. The Selected attributes are 3,5,6,9,10 : 5(NM, NAgg,NDep,NGenH,MaxHagg) are considered as the selected First Input matrix from Weka. In second set of evaluation, there are six parameters selected NM, NAssoc, NAgg, NDep, MAxHagg, MaxDIT and output genero model (class parameter) respectively.

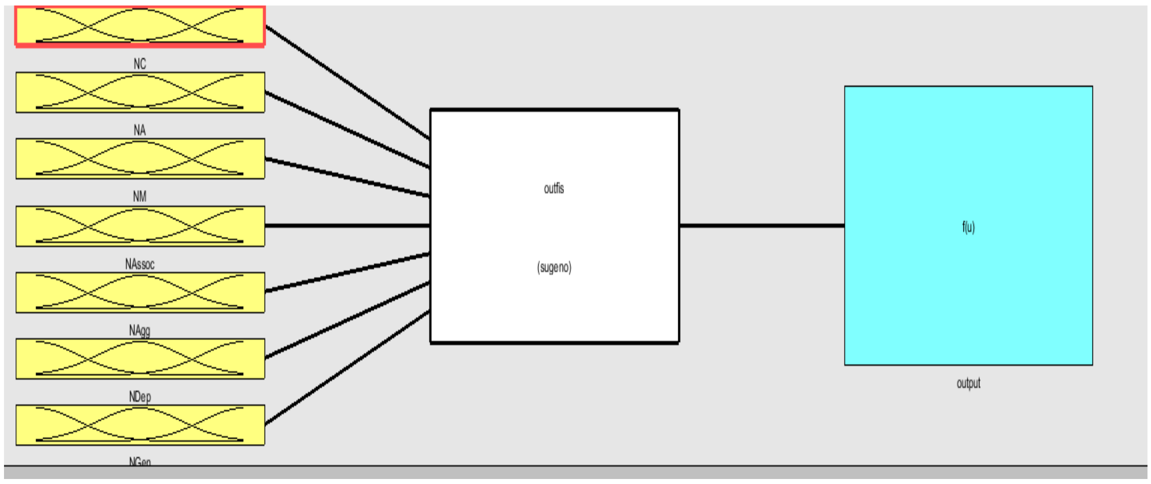

Figure5. ANFIS Model with the 7 Parameters

C. *Attribute Selection with the 7 Parameters.*

The following parameters were selected from the given 12 input parameters from 28 class diagrams. It uses the full data set.

The selected attributes are NC, NA, NM, NAssoc, NAgg, NDep, NGen and output genero model (class parameter) respectively.

TABLE II TRAINING DATASET (6 PARAMETERS)

TABLE III. TESTING DATASET (6 PARAMETERS)

S.No	NM	NAssoc	NAgg	NDep	Max Hagg	Max DIT	**genero model**	NM	NAssoc	NAgg	NDep	Max Hagg	Max DIT	**genero model**
1	30	0	0	0	0	3	**6.9**	8	1	0	0	0	0	**10.0**
2	38	0	0	0	0	4	**7.9**	12	1	1	0	1	0	**8.8**
3	76	10	6	2	2	2	**5.6**	15	1	2	0	2	0	**8.8**
4	88	10	6	2	4	3	**3.9**	12	3	0	0	0	0	**8.8**
5	94	6	6	1	4	4	**1.5**	21	1	3	0	2	0	**8.8**
6	98	12	7	3	4	4	**0.0**	12	1	0	0	0	0	**8.6**
7	47	1	5	0	4	1	**5.8**	12	2	0	0	0	0	**7.5**
8	65	3	5	0	3	4	**5.1**	14	3	2	0	2	0	**5.8**
9	79	11	6	0	4	3	**3.9**	12	2	0	1	0	0	**8.8**
10	69	1	5	0	2	5	**5.1**	20	1	3	0	2	1	**5.8**
11	73	9	7	2	4	1	**4.4**	26	2	3	0	3	1	**5.8**
12	84	14	4	4	2	3	**1.5**	37	3	3	0	3	1	**5.8**
13	77	4	9	0	3	4	**2.3**	35	3	2	1	2	1	**5.8**
14	47	6	6	0	2	2	**7.9**	26	0	0	0	0	2	**8.8**

TABLE I V TRAINING DATA SET WITH THE 7 PARAMETERS

S.No	NC	NA	NM	NAssoc	NAgg	NDep	NGen	Output Class
1	6	10	14	3	2	0	2	6
2	3	9	12	2	0	1	0	9
3	7	14	20	1	3	0	2	6
4	9	18	26	2	3	0	4	6
5	7	18	37	3	3	0	2	6
6	8	22	35	3	2	1	2	6
7	5	9	26	0	0	0	4	9
8	8	12	30	0	0	0	10	7
9	11	17	38	0	0	0	18	8
10	20	42	76	10	6	2	10	6
11	23	41	88	10	6	2	16	4
12	21	45	94	6	6	1	20	2
13	29	56	98	12	7	3	24	0
14	9	28	47	1	5	0	2	6

TABLE V TEST DATA SET WITH THE 7 PARAMETERS

S.No	NC	NA	NM	NAssoc	NAgg	NDep	NGen	Output class
1	2	4	8	1	0	0	0	10
2	3	6	12	1	1	0	0	9
3	4	9	15	1	2	0	0	9
4	3	7	12	3	0	0	0	9
5	5	14	21	1	3	0	0	9
6	3	6	12	1	0	0	0	9
7	4	8	12	2	0	0	0	8
8	18	30	65	3	5	0	19	5
9	26	44	79	11	6	0	21	4
10	17	32	69	1	5	0	19	5
11	23	50	73	9	7	2	11	4
12	22	42	84	14	4	4	16	2
13	14	34	77	4	9	0	7	2
14	17	34	47	6	6	0	11	8

Both the predicting models are evaluated by using the test data set. The evaluating process is done by two classes, one is easy to maintain and another one is the difficult to maintain the software. The maintainability value is evaluated from the trained SFIS. It is the range between 0 and 12. The value from 0-5 is considered as the *easy to maintain* and the value from 6-12 as the *difficult to maintain*. From the predicted model the error values are evaluated. The output is considered as error, when the model is misclassified the output as *easy to maintain* as *difficult to maintain*. It is extremely well to the software developers for the decision making regarding the cost of the maintainability.

5.4. Experimental Results

Generally, CFS-ANFIS can exactly formulate the 2 different predicting models for measuring the maintainability of the software from the UML class diagram. Learning algorithm, ANFIS can learn and create the learning rules from the input data and a target attribute is set up for each predicting model. In addition, a target attribute is possibly occurred for 2 values which are 0 (encountering the difficult) and 1 (normal easy state). This study shows that CFS-ANFIS creates 2187 learning rules from the input data, 7selected attributes and 1Target attribute, and also ANFIS creates 4 learning rules from the input data, 5 selected attribute and an additional one target attribute.

5.4.1. FIS Creation and Estimation of the Maintainability

The ANFIS network was trained with the given input training data set and output. The ANFIS generates its own SFIS with the given input and output pairs and generates a set of rules. It aims to get the minimum total sum of mean squared errors from the measured target value. The whole training data set which includes the seven input parameters and one output parameter will be in the network for some iteration to be happened to reduce the error value. Once the network learns from the training data set, the error value falls to zero.

The hybrid learning algorithm was used in its ANFIS architecture and the training was based on 30 to 100 epochs. The initial step is the preparation of the training data to train with the ANFIS model in MAT lab. It takes the input as a matrix form, where the last column in the matrix is represented as the output parameter and the other previous columns represent the input parameter to the ANFIS system.

To generate the initial FIS, the fuzzy tool box is used in MAT Lab. It provides a situation to generate the FIS with the given number of input parameters, output parameter and the membership function. It also generates the rule from the training data set. Three triangular typed fuzzy MFs are used per each input. The MFS are selected to describe the input and output variables. There are 9 rules for each one regarding the two inputs with fuzzy sets to specify the output. The ANFIS structure with the 7 input parameters and one output are shown in the figure 5.The MFS for the input parameters are shown in figure 6.1 to figure6.7

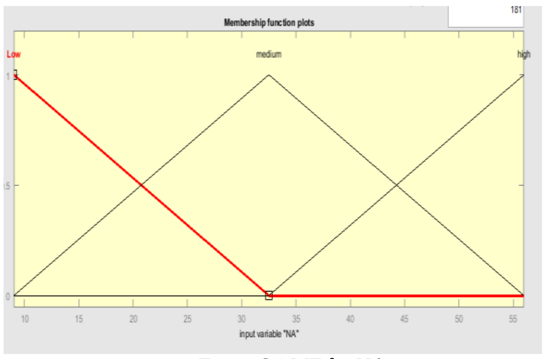

Figure 6.1 MF for NA

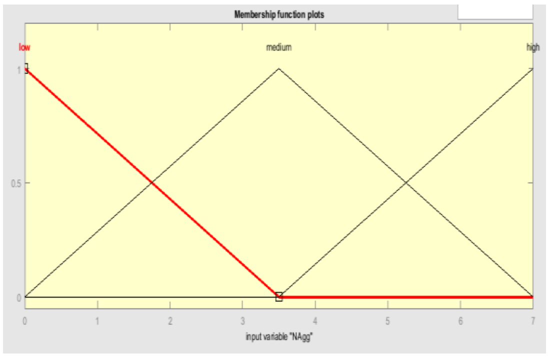

Figure 6.2 MF for NAgg

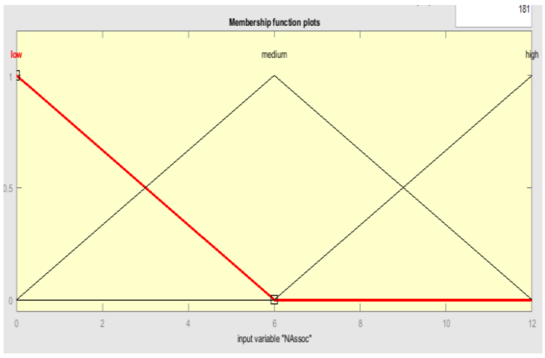

Figure 6.3 MF for NAssoc

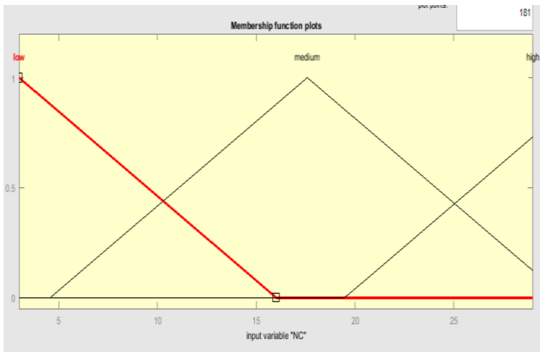

Figure 6.4 MF for NC

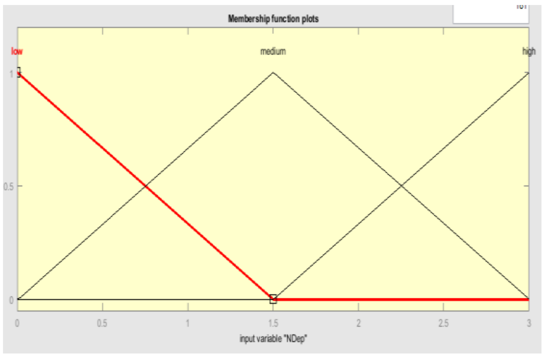

Figure 6.5 MF for NDep

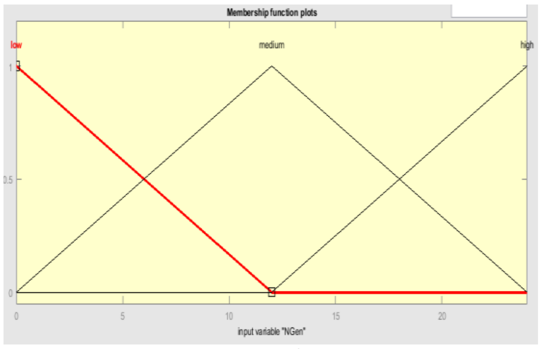

Figure 6.6 MF for NGen

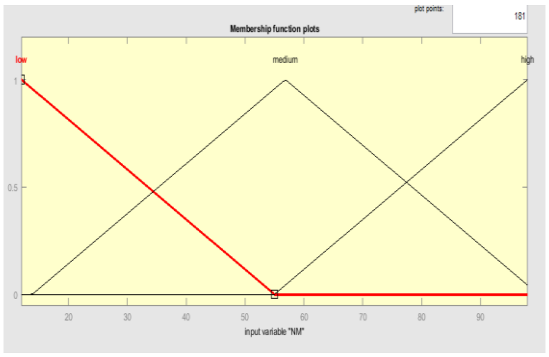

Figure 6.7 MF for NM

VI. RESULT ANALYSIS

After generating the predicting model, the test dataset is provided to the model to evaluate the error. The Table VI discusses the training and testing dataset with different epochs. In terms of the prediction for CFS-ANFIS which archives 0.00032 for the training and 4.9 for the testing data set. In contrast, the second model with 6 parameters has got 0.00038 for the training and 4.19 for the testing. Surprisingly, all the errors which are found in the 5 parameters CFS-ANFIS are high. On the other hand, it is shown that accuracy of the estimation of CFS-ANFIS (7) model is higher than CFS-ANFIS (5) model. In addition the error rate of the CFS-ANFIS (6) is lower than CFS-ANFIS (7). It is clearly pointed that the CFS-ANFIS worked better with the seven parameters than the CFS-ANFIS with 5 parameters for the estimation of the maintainability of the software from the UML class diagram.

TABLE VI TESTING AND TRAINING ERROR

FIS generation method	No. of mf	Mf type	Epoch	Training Error	Testing Error
Grid Partitioning method with the 7 parameters	3*3*3*3*3*3*3	Trimf	30	0.0117	6.8
			40	0.00032	5.4
			50	0.00032	5.0
			80	0.00032	4.9
			100	0.00032	4.9
Grid Partitioning method with the 6 parameters	3*3*3*3*3*3*3	Trimf	10	0.000394	4.19
			20	0.000393	4.19
			30	0.000393	4.19
			40	0.000392	4.19

			50	0.000392	4.19
			60	0.000390	4.19
			80	0.000390	4.19
			100	0.000389	4.19
			150	0.000388	4.19
			200	0.000384	4.19

A. Training Error with different epochs

The result shows the average testing error and training error of the model with various epochs. The training error with epoch are presented in the figure 7.1 to 7.3.As the training of different epoch the error rated are decreased and it became a constant in the error level in the figure 7.3.

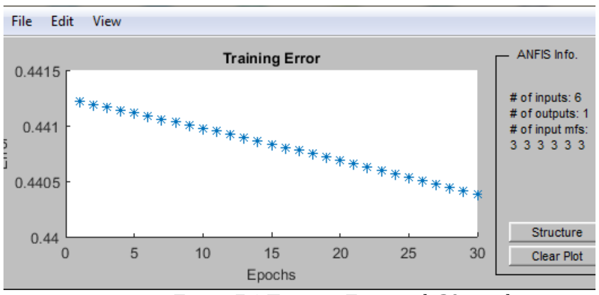

Figure 7.1 Training Error with 30 epoch

Training error with 30 epoch are presented in the figure 7.1.As the training epoch increases the error value is decreased to 0.05 and further it reduces from 0.44 to 0.43 in the 40 epoch are shown in the figure 7.2.

The training error with 80 epoch are presented in the figure 7.3.As the training epochs are increased the error rated shown in the figure7.3 are constant or it has achieved a steady state in error level.

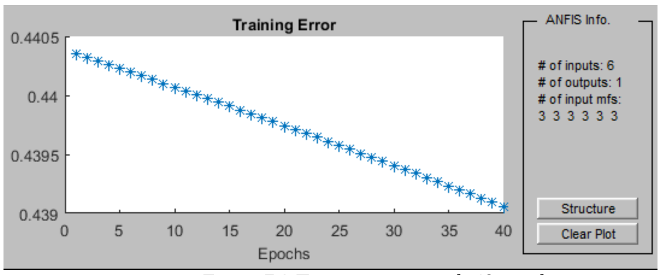

Figure 7.2 Training errors with 40 epoch

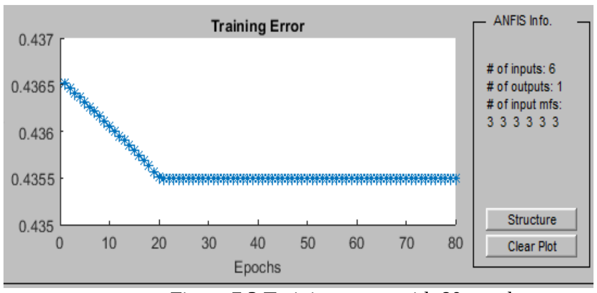

Figure 7.3 Training errors with 80 epoch

Further the training of the same, SFIS does not improve the error level. As an increase in the epoch there is no change in the performance. Thus, there is an average decrease in error rate and then it stops it's tuning of the model.

B. Model validation

The predicted model was evaluated its performance by comparing the measured value of the maintainability with the actual of the maintainability .The table VII shows the maintainability value of the predicted value with the 7 parameters and 5 parameters.

TABLE VII PREDICATED AND ACTUAL VALUE OF THE MAINTAINABILITY

	With the 7 parameter				With the 5 parameter			
S.No	**Actual value**	**A Output class**	**M Value**	**M Output class**	**Actual output**	**A output class**	**M output**	**M output class**
1	10	E	12	E	10	E	4	D
2	9	E	10	E	9	E	5	E
3	9	E	7	E	9	E	5	E
4	9	E	9	E	9	E	5	E
5	9	E	6	E	9	E	4	D
6	9	E	12	E	9	E	5	E
7	8	E	10	E	8	E	5	E
8	5	D	1	D	6	D	4	D
9	4	D	0	D	9	E	5	E
10	5	D	1	D	6	D	4	D
11	4	D	5	D	6	D	3	D
12	2	D	2	D	6	D	5	E
13	2	D	1	D	6	D	6	E
14	6	E	1	D	9	E	6	E

E-Easy to maintain D-Difficult to maintain

C. Evaluating the output using the Curve fitting tool

A curve fitting tool is used to measure performance of the predicted and actual value of the maintainability in terms of R^2, adj R^2 and RMSE value. As the maintainability value measurement with 7 parameters has been achieved the low error value .To measure the performance, this model is considered. Here, the X axis represents the actual value whereas the Y axis represents the predicted output of the maintainability and the performance factors measured from the graph are given in the Table VIII.

TABLE VIII PERFORMANCE FACTOR

Performance Factor	Value for 7 parameter	Value for 5 Parameter
SSE:	0.7696	SSE: 29.37
R-square:	0.9359	R-square: 0.05036
Adjusted R-square:	0.9305	Adjusted R-square: -0.02878
RMSE:	0.253	RMSE: 1.564

D. Performance measures for the classification

The correctness of a classification can be evaluated by computing the number of correctly predicted class, examples, *easy maintain as easy* (true positives), the number of correctly recognized examples that do not belong to the same class, *difficult to maintain* as *difficult* (true negatives), and either was incorrectly assigned to the class (false positives) or these were not recognized as class examples (false negatives). These four counts create a confusion matrix shown in the Table IX for the case of the **binary classification** and Table X shows the measure of binary classification.

TABLE IX CONFUSION MATRIX FOR BINARY CLASSIFICATION

True positive (TP)	**False positive** (FP)
False negative (FN)	**True negative** (TN)

TABLE X MEASURES FOR BINARY CLASSIFICATION RESULTS

Experimental Set Up	Accuracy	Sensitivity	Specificity	Precision
With the 7 Parameter	0.9286	0.8750	1	1
With the 6 Parameter	0.7857	1	0.7000	0.5714
With the 5 Parameter	0.7143	0.6000	0.7778	0.6000

CONCLUSION

This paper presents a new approach with the machine learning system with CFS. The CFS is used to reduce the number of features to get a better prediction model. It is incorporated with ANFIS model to produce the best trained model. The CFS reduces the higher level of dimensionality of the data for both discrete and continuous form. According to the objective of the study, this paper highlights on the comparison of the attribute selection techniques which are CFS in terms of different parameters. Furthermore, these techniques can appropriately associate with the ANFIS for estimating the maintainability of the software. Finally, the CFS-ANFIS has the lower error rate of prediction than the ANFIS with the 5 parameters .Therefore, these parameters impact more on the maintainability of the software. So, it recommends the software developers to concern these parameters more in the design phase. It is highly commanded to use the CFS-ANFIS for predicting, measuring the maintainability of the software to select optimized parameters

References

[1] "ISO/IEC 14764:2006 Software Engineering —Software Life Cycle Processes — Maintenance". Iso.org. 2011-12-17.Retrieved 2013-12-02.

[2] Jang, J.S.R.(1992).Neuro fuzzy modeling: architecture, analyses and applications. Unpublished PhD Dissertation, Department of Electrical Engineering and Computer Science, University of California, Berkeley, California.

[3] Jang J.S.R. (1993).ANFIS: adaptive network based fuzzy inference system. IEEE Transaction on Systems, Man and Cybernetics,23(3),665–685.

[4] Mordal,K., Nicolas Anquetil, Jannik Laval , Alexander Serebrenik , BogdanVasilescu and StéphaneDucasse,(2012). Software quality metrics aggregation in industry.*Journal of Software: Evolution and Process, 25(10), 1117-1135.*

[5] Ahmed, M., & Al-Jamimi, H.A. (2013).Machine learning approaches for predicting software maintainability: a fuzzy-based transparent model. *Special Issue on Empirical Studies in Software Engineering, IET Software*,7(6),317-326.

[6] Al-Jamimi, H.A ,& Ahmed, M.(2013.) Machine learning-based software quality prediction models: state of the art. *Proceedings of Fourth International Conference on Information Science and Applications* (pp.1-4), Pattaya,Thailand.

[7] Genero, M., Piattini, M., &Calero, C. (2005).A survey of metrics for UML class diagrams.*Journal of Object Technology,* 4(9), 59-92.

[8] Genero, M., Olivas, J., Piattini, M., & Romero, F.(2001).Using metrics to predict OO information systems maintainability. *In Proceedings of 13th International Conference on Advanced Information Systems Engineering (CAiSE 2001), Lecture Notes in Computer Science,* 2068, 2001, pp.388-401.Interlaken, Switzerland.

[9] Genero, M., Piattini, M., Manso, E. and Cantone, G.(2003).Building UML class diagram maintainability prediction models based on early metrics.InProc. *Ninth International Software Metrics Symposium (*pp. 263-275).

[10] Genero, M., Piatini, M., &Manso, E. (2004). Finding “early” indicators of UML class diagrams understandability and modifiability, In Proc. 2004 *International Symposium on Empirical Software Engineering* (pp. 207-216).

[11] BogdanDit, Andrew Holtzhauer, Denys Poshyvanyk,&HuzefaKagdi, (2013). A dataset from change history to support evaluation of software maintenance tasks.In Mining Software Repositories (MSR).*10th IEEE Working Conference on. 2013*. IEEE.

[12] Chai, T., &Draxler, R.(2014).Root mean square error (RMSE) or mean absolute error (MAE)? Geoscientific Model Development Discussions, 7, 1525-1534.

[13] Bennett, K.H., &Rajlich, V.T.(2000). Software maintenance and evolution: a roadmap. *In Proceedings of the Conference on the Future of Software Engineering*.ACM.

[14] Dzidek, W.J., Arisholm, E., & Briand, L.C. (2008).A realistic empirical evaluation of the costs and benefits of UML in software maintenance. *IEEE Trans.on Software Engineering*,34(3),407-432.

[15] McCall, J.A., Richards, P.K. & Walters G.F.(1977).Factors in software quality. Concepts and Definitions of software quality. DTIC Document

[16] www.Coefficient of Determination (R-Squared) - MATLAB &

Simulink - MathWorks India.html

[17] W.-H. Ho, J.-T. Tsai, B.-T.Lin, and J.-H. Chou, ``Adaptive network-based fuzzy inference system for prediction of surface roughness in end milling process using hybrid Taguchi-genetic learning algorithm," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 3216_3222, 2009.

[18] P.R. Therasa and P. Vivekanandan, “Adaptive Neuro-Fuzzy Inference System for Assessing the Maintainability of the Software” 2017 Ninth International Conference on Advanced Computing (ICoAC) Dec 2017.

[19] Marcela Genero, M. Esperanza Manso, Corrado Aaron Visaggio, Gerardo Canfora, Mario Piattini.(2007).Building measure-based prediction models for UML class diagram maintainability. Empirical Software Engineering 12(5): 517-549.

[20] Yi, T. & Wu, F. (2004).Empirical Analysis of Entropy Distance Metric for UML Class Diagrams. ACM SIGSOFT Software Engineering Notes, 1-6.

[21] Yi, T., & Wu, F. (2010).Comparison research of two typical UML-class-diagram metrics: Experimental software engineering. In Proc. 2010 International Conference on Computer Application and System Modeling (pp. 86-90).

[22] Yao Lu, Xinjun Mao, &Zude Li (2016).Assessing Software Maintainability Based on Class Diagram Design: A Preliminary Case Study,,Lecture Notes on Software Engineering, 4(1).

[23] Zhou, Y., &Xu, B. (2008).Predicting the maintainability of open source software using design metrics. Wuhan University Journal of Natural Sciences, 13(1), 14-20.

[24] Zhou, Y., &Xu, B. (2003).Measuring structure complexity of UML class diagrams. Journal of Electronics (China), 20(3), 227-231.

[25] Alshayeb, M. (2013).On the relationship of class stability and maintainability. IET Software,339-347.

[26] D. Kozlov et al., “Assessing maintainability changes over multiple software releases”, Journal of Software Maintenance and Evolution: Research and Practice, 2008. 20(1):pp. 31-58.

[27] Aggarwal KK, Singh Y, Chhabra JK. An integrated measure of software maintainability.Annual Proceedings on Reliability and Maintainability Symposium. IEEE Computer Society Press: Silver Spring MD, 2002; 235–241

[28] Laitinen K. Estimating understandability of software documents. ACM SIGSOIT, vol. 21, 1996; 81–92.

[29] KulthonKasemsan and WonlopBuachoom.“The Comparative of Attribute Selection Techniques between CFS and Consistency by Using ANFIS for Thai Enterprises Bankruptcy Prediction”.Journal of Information Science and Technology.Vol.1.Issue 1.Jan-Jun 2010.

[30] Hall, M. (1999). Correlation based feature selection for machine learning. Doctoral dissertation,Universityof Waikato, Dept. of Computer Science.

[31] WEKA,http://www.cs.waikato.ac.nz/ml/weka, Last access, 8 April 2008.

[32] Zhou, Y., &Xu, B. (2008).Predicting the maintainability of open source software using design metrics. *Wuhan University Journal of Natural Sciences*, 13(1), 14-20.