International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 6, Issue 5, September- October 2017 ISSN 2278-6856

Code Clone Detection Using COCOMO-1

Ekta Manhas[1] , Samriti Rana[2]

[1]Research scholar (CSE)
Rayat Group Of Institutions, Railmajra, Punjab, India

[2]Associate Professor (CSE)
Rayat Group Of Institutions, Railmajra, Punjab, India

Abstract: ***Now adays, Copy and Paste of code fragments has been regularly practiced in development of software. Because of limitations of time and lack of knowledge, the programmers use the code strategy known as code cloning. Clones may cause many problems that are the probability of errors and the maintenance cost get increased. The modifications become difficult because of clones. Therefore, the detection and removal of clones is necessary. It has been observed that lot of tools, techniques and classifiers have been already tried in the concept of textual parameter code cloning detection, but there are chances of improvement of the accuracy pattern of classification in code cloning. So, the motive of this research work is to enhance the accuracy of the system with reduction of error rate using neural network classifier. In the proposed work, ANN would be used that involves training of data. Utilization of ANN has been done for code cloning as it provides good training tool with robust results. The analysis has been carried out in MATLAB environment and the metrics like accuracy and error rate are calculated. The accuracy upto 95 % have been obtained.***

Keywords: Code clone, COCOMO-II, Neural Network, Accuracy, Error rate

1. INTRODUCTION

Cloning of code has become one of the easiest ways to complete a project for a programmer without investing effort and time for programming section. It's a loss for a programmer who really works hard in writing code for a project [1]. Clone detection is concerned with finding similar pattern in source code, interpreting and using them in design, testing and other software engineering problems. Code clones may adversely affect the software systems' quality, especially their maintainability and comprehensibility. It would be interesting to see how the artificial intelligence works with the clone detection.

Code cloning can be defined as an act of reusing a segment of code by copying it from one section of the software and then pasting it with or without some slight alterations into another section of the software. Most of the clone detection tools give results as clone pairs or clone classes. Two code segments form a clone pair, if codes are related to each other by an equivalence relation [2]. An equivalence relation holds all reflexive, symmetric and transitive relations. A clone pair is defined as a pair of matching code segments. Clone class is defined as a set of code segments with similar code portions. Each code segment in a clone class forms a clone pair with other code segments of that class [3].

On the basis of program text similarity, code clones can be classified into 3 types [4]:

i. Type-1 Clones

If a code segment is copied with minor amendments in whitespaces, layout and comments then it comes under type-1 or exact clones.

ii. Type-2 Clones

If a code segment is copied with some amendments in name of the variables, functions, types and identifiers.

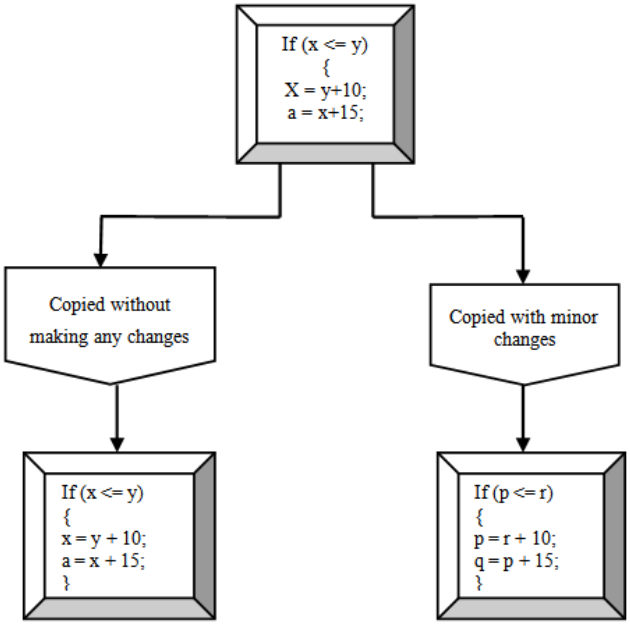

Figure 1 Code clone example

iii. Type-3 Clones

If a code segment is copied with some changes like insertion or deletion of statements along with change in name of variables, functions and type, then it comes under type-3 or near miss clones.

iv. Functional similarity

If two code segments perform the same functionality but codes are having different syntax, then codes are said to be type-4 or semantic clones. These clones are the most difficult to detect. In case of semantic clones, it is not

necessary that the code clone is a copy of any other code segment.

2. PROPOSED TECHNIQUES

The aim of this research is to create a framework which includes cost model as comparison metrics. The framework also adds up a classification module over the cost model [5]. The cost model has additive parameters of MBD to make the framework stronger. If the classified parameters of original and tested code are not varying by more than 20-25 percent, then the test code would be considered as copied code. The problem statement of this research work also includes design and development of a clone management system. A managed clone system prevents the architecture from any extra effort and saves a lot of time [6].
The objective of the technique is to reduce the error rate by enhancing the accuracy rate using machine learning algorithms. Then, the result validation is being done using FAR (False Acceptance Ratio), FRR (False Rejection Rate), F-Measure and accuracy in MATLAB 2010a environment [7] [8].

2.1 COCOMO –I (Constructive Cost Estimation Model)

The COCOMO [7] (Constructive Cost Estimation Model) is proposed by DR. Berry Boehm in 1981 and that's why it is also known as COCOMO'81. It is a method for evaluating the cost of a software package [9]. According to him software cost estimation should be done through three stages:

i. Basic COCOMO Model
ii. Intermediate COCOMO Model
iii. Complete/Detailed COCOMO Model

COCOMO-I models depend upon the two main equations [10]:

a. Development Effort

$$MM = a * KDSI^b \quad (1)$$

Which is based on MM - man-month / person month / staff-month is one month of effort by one person. Note: In COCOMO'81, there are 152 hours per Person month. According to organization this values may differ from the standard by 10% to 20%.

b. Efforts and Development Time (TDEV)

$$TDEV = 2.5 * MM^c \quad (2)$$

Note: The coefficients a, b and c depend on the mode of the development.

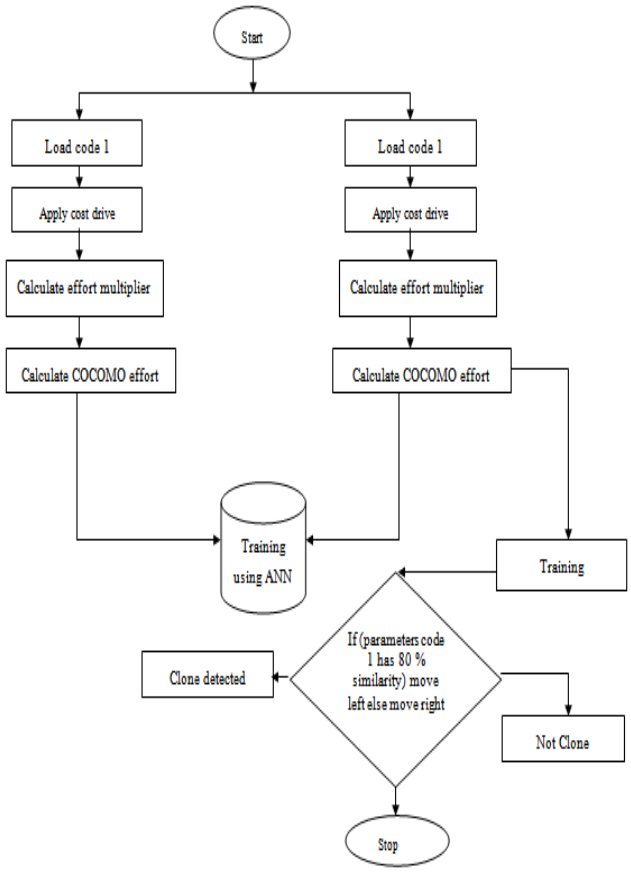

Figure 2 Simulation Model

2.2 Neural Network

Neural network is one of the important classification algorithms. It generally consists of various numbers of neurons called units that are arranged in the form of layers [11]. It is basically used for training as well as testing. BPNN (Back propagation neural network) is used for solving many NP problems. A typical neural network consists of various numbers of neurons called units that are arranged in form of layers and each of which is connected to next layer via layers. Neural network mainly used for training. Training can be either supervised or unsupervised. In supervised training, system learns by trying to predict outcomes for known examples [12]. System compares its predictions with the known results and learns from is mistakes. In unsupervised training, system with no output or result is shown as part of training process. Neural network is consisted of three layers.

i. **Input layer:** The trained data is provided on this layer.
ii. **Hidden Layer:** Processing of the trained data is done in this layer.
iii. **Output Layer:** Classified results are taken from this layer.

3. SIMULATION MODEL

This research has dealt with the study and analyzing various methods of code clone detection also to study

neural network algorithm in order to check the suitable one for the clone detection. A framework has been developed which includes cost model as parameters with addition of hybrid token and metrics based model. A detected clone management has been detected for identifying the clones more easily.

The steps undertaken to simulate the work are defined below:

Step: 1 Start
Step: 2 Select the First Code File
Step: 3 Read text file
Step: 4 Save code 1 file
Step: 5 Select the Second Code File
Step: 6 Read text file
Step: 7 Save code 2 file
Step: 8 load Code1Data.mat
Step: 9 Apply cost model for both the codes
Step: 10 Determine effort multiplier along with COCOMO 1 effort for both the codes
Step: 11 Training has been performed by using neural network and then values are stored in the database.
Step: 12 Testing of code 2 has been performed in order to check the clone.
Step: 13 Step: 13 If the test data is similar to 80 % of the saved data then the code is treated as clone otherwise the code is treated as original code

4. SIMULATION RESULTS

This section explains the result obtained after the implementation of the first two objectives that is to study and understand the different methods of code clone detection with neural network algorithm in order to check the suitable one for the clone detection and to develop a framework which includes cost model as parameters with addition of hybrid token and metrics based model.

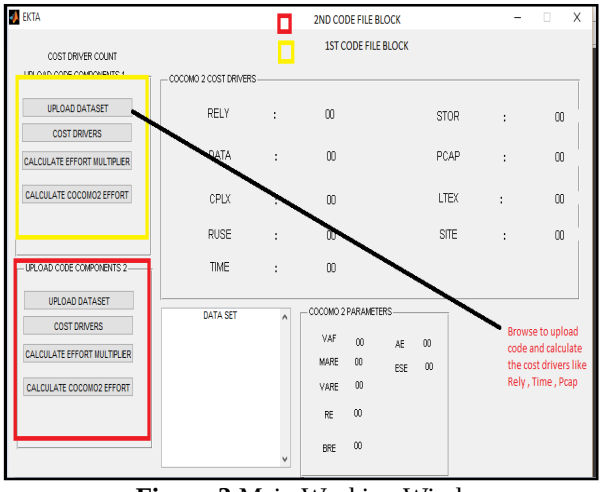

Figure 3 Main Working Windows

Main working window with the ease of controls is shown above. The concept is to evaluate the code components like rely, data, complexity and other factors and with the help of the evaluated components, effort has been calculated. The collected components would be passed to a classifier in order to match the components co relation.

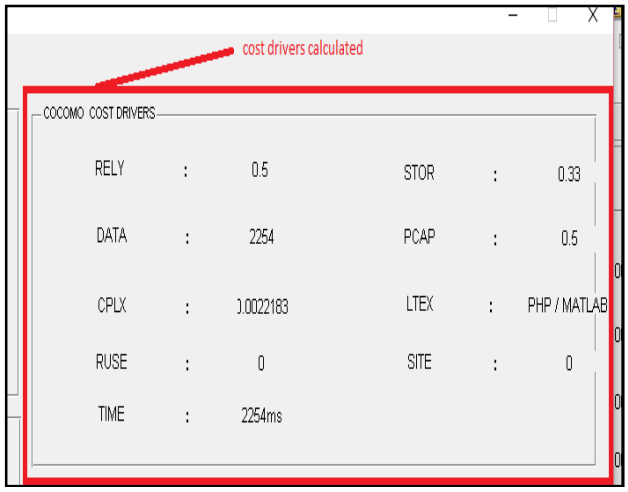

Figure 4 Calculation of cost drivers

The components would be calculated for both the codes and the training module would be prepared. The classification accuracy would predict that how much code is copied in both the contents.

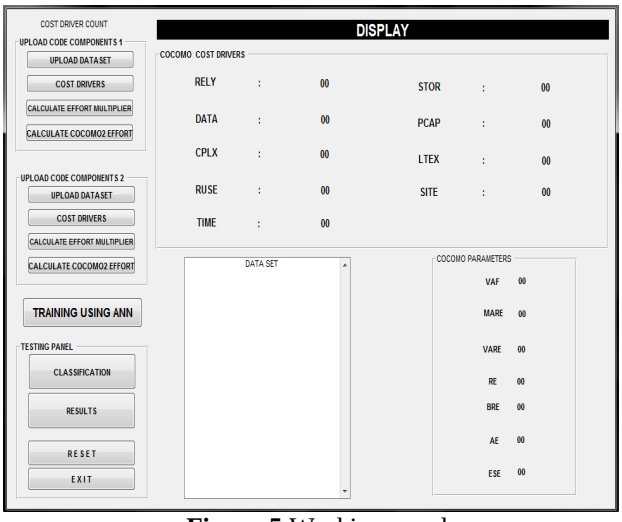

Figure 5 Working panel

The above figure represents the working panel of proposed work regarding code clone detection using ANN. The figure is comprised of 3 different parts on the left side. The first part is to upload the base code, whereas, the second part is for uploading the code for which the clone detection has to be performed. The third part classifies whether the code to be tested is original or is a clone of the base Code.

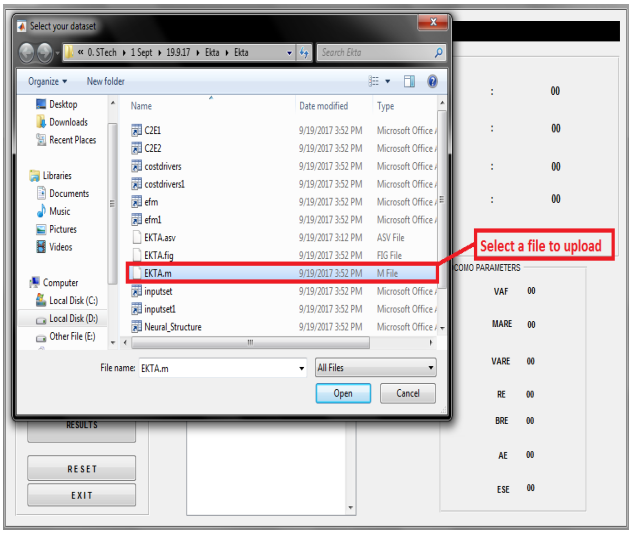

Figure 6 Uploaded data

The above figure depicts the uploading of the base code, from which the other code is matched in order to detect whether it is cloned or not. After clicking the upload dataset button the wizard appears as shown in the above figure, then locate the code in your device's storage and after selecting the appropriate code click the open button on the wizard window to complete the uploading process.

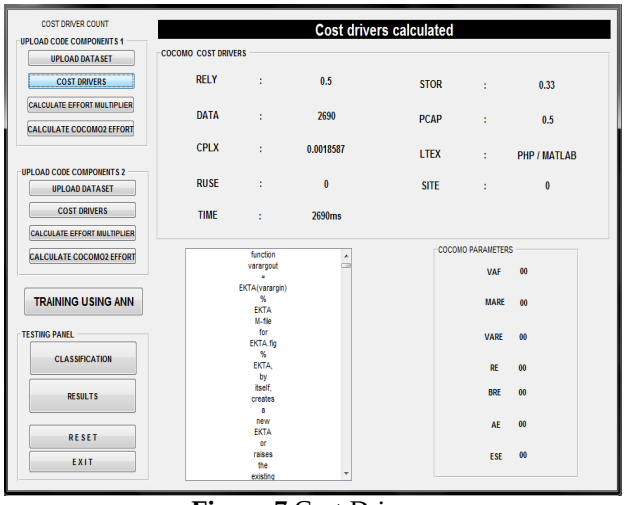

Figure 7 Cost Drivers

The above figure represents the next step after the uploading of the first code. The next step includes calculating the cost drivers which is performed by pressing the button lying below the upload dataset button.

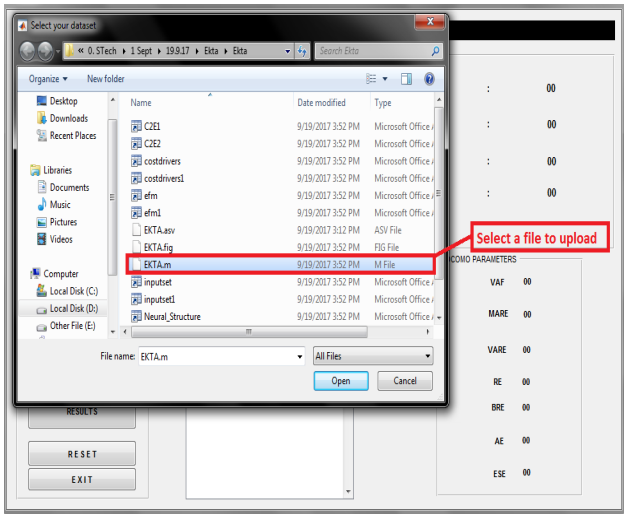

Figure 8 Loading 2nd code

Here, in the figure shown above the uploading of the second code which has to be analyzed to detect the clone is demonstrated. Press upload button on the second part of the working panel on the left for uploading the second code. Again the wizard window appears on which the code has to be located and after selecting the appropriate code file, the Open button on the wizard window is pressed in order to finalize the uploading process.

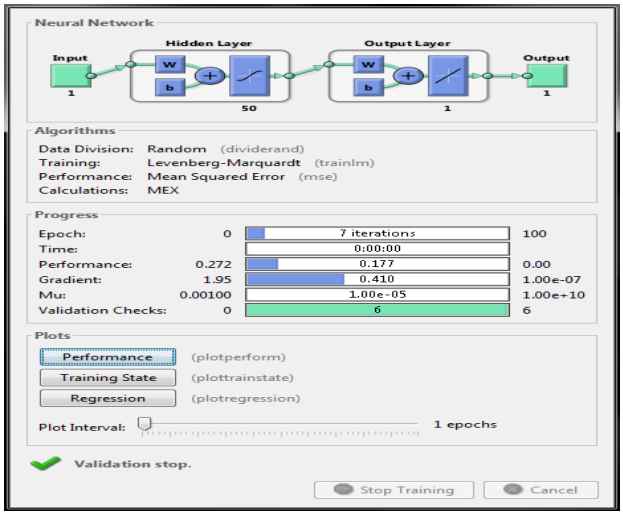

Figure 9 Training neural network

The numbers of maximum epochs taken are 100 out of which 7 iterations were converged. Performance, Training as well as the regression states are considered while the training of neural network takes place. From the 6 validation checks, 6 validation checks are considered.

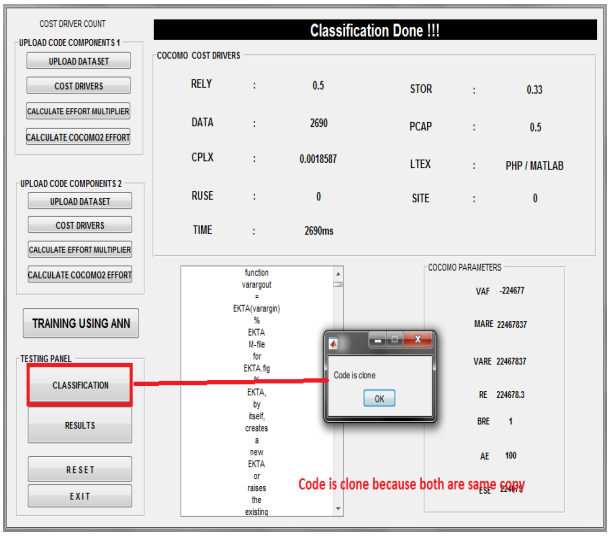

Figure 10 Classification

In the above figure the classification button on the third part at the bottom-left of the working panel is presented. This button classifies the datasets and check whether both the uploaded code is same or not. I.e. detecting whether the second code is same copy or duplicate copy of the base code or not.

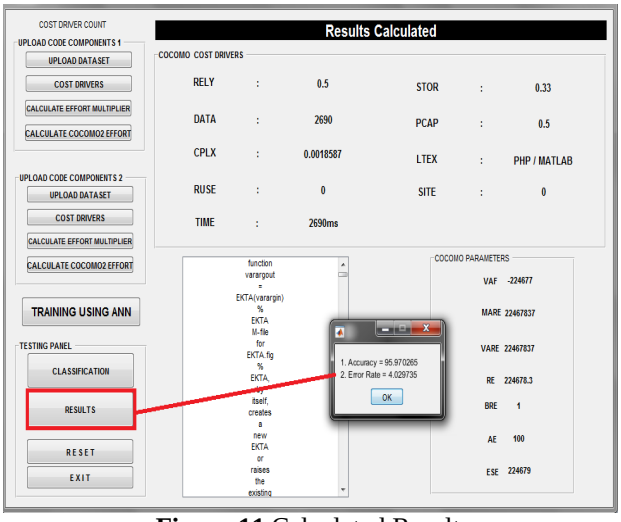

Figure 11 Calculated Result

The figure displayed above represents the result window. The results of the proposed work regarding the detection of code clone using ANN is carried out by pressing the Results button underneath the Classification button presented in the third part at the left-bottom of the working panel. This button reveals the accuracy of the proposed work and error rate calculated.

5. CONCLUSION AND FUTURE SCOPE

Code cloning is a term used to mention that some part of the pre-existing code has been utilised. The obvious danger of cloning is that it increases the maintenance cost as well as increases the complexity of the code. Cloning has become fundamental need of industry because of its main advantage that is cost. That is the reason why code cloning has been widely utilized as a part of vast software businesses. So to identify clones, robust and accurate methods are needed at urgent need. Numerous methodologies have been created to distinguish clones with different accuracy rates. In this research, usage of metric based methods was done. This report shows the implementation with neural network technique. From result evaluation accuracy of 95.97% has been achieved along with the error rate of 4.029.

In future for detecting clone and to increase the accuracy of the system other classification techniques named as SVM can be used in hybridization with artificial neural network.

REFERENCES

[1] F. C alefato. F. Lanubile and T. Mallardo, “Function Clone Detection in Web Applications: A Semi automated Approach”, in Journal of Web Engineering. vol. 3. number 1, 2004: 3-21

[2] H. Liu. Z. Ma. L. Zhang and W. Shao, “Detecting duplications in sequence diagrams based on suffix trees”, in Proceedings of IPSE C, 2006: 269-276.

[3] Ira Baxter, Andrew Yahin Leonardo Moura, Marcelo Sant Anna, “Clone Detection Using Abstract Syntax Trees”, in Proceedings of the 14th International Conference on Software Maintenance tIC$M'98), Bethesda. Marjdand, November 1998: 368-377

[4] IstvanSiket& Rudolf Ferenc, “Calculating Metrics from Large C-H- Programs”, in 6th International Conference on Applied Informatics Eger, Hungarv,January,2001: 27-31

[5] Roy, Chanchal K., James R. Cordy, and Rainer Koschke. “Comparison and evaluation of code clone detection techniques and tools: A qualitative approach.” Science of computer programming 74.7 (2009): 470-495.

[6] Kevin Greenan, “Method-Level Code Clone Detection on Transformed Abstract Syntax Trees using Sequence Matching Algorithms”, Student Report. University of California - Santa Cruz. Winter 2005

[7] Lingxiao Jiang, GhassanMisherghi,Zhendong Su. and Stephane Glondu, “DECKARD: Scalable and Accurate Tree-based Detection of Code Clones”, In Proceedings of the 39th International Conference on Software Engineering IICSE '07), Minnesota, USA, May 2007: 96-105.

[8] Kodhai, S. Kanmani, A. Kamatchi, R. Radhika and B. V. Saranya. “Detection of Type-1 and Type-2 Code Clone using Textual Analysis and Metrics” inProceedings of the 2010 International Conference on Recent Trends in Information, Telecommunication and Computing, Kerala, 2010:241-243.

[9] M. A. Yahya, R. Ahmad and S. P. Lee, “Effects of software process maturity on COCOMO II’s effort estimation from CMMI perspective,” *2008 ieee international conference on research, innovation and vision for the future in computing*

and communication technologies, Ho Chi Minh City, 2008, pp. 255-262.

[10] Wang, Sun-Chong. "Artificial neural network." Interdisciplinary computing in java programming. Springer US, 2003. 81-100.

[11] M. Madheswaran and D. Sivakumar, "Enhancement of prediction accuracy in COCOMO model for software project using neural network," *Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, Hefei, 2014, pp. 1-5.

[12] W. Yue, Z. Biaobiao; L. Jiabin; K. L. Du "Fuzzy Logic and Neuro-fuzzy Systems: A Systematic Introduction", International Journal of Artificial Intelligence and Expert Systems (IJAE), Volume (2) : Issue (2),2011.