International Journal of Engineering & Technology, 7 (3.8) (2018) 117-124

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Fuzzy Logic Control Neural Network Hybrid System for Identification, Classification of Software Reusability Components through Relationships of Lattice Factors

T.Rajani Devi[1]*, B.Rama[2]

[1]*Computer Science Department, University College, Kakatiya University, Warangal, Telangana State, India*, rajanireddy.phd@gmail.com
[2]*Computer Science Department, University College, Kakatiya University, Warangal, Telangana State, India*. rama.abbidi@gmail.com
**Corresponding author E-mail:* rajanireddy.phd@gmail.com

Abstract

The software reusability mode is highly required field for successful execution of artificial intelligence, machine learning based applications to fulfill the present and future human needs. The identification, classification and measuring the required components are key-roles concerns in fast development of software reusability components for producing the high quality software. This paper is proposing the Fuzzy Logic Controller Neural Network Hybrid System which is implicated to recognize the affecting factors of component reusability execution by instituting the strong, week relationships in among these considered factors to fulfill the user requirement. This approach considered eleven effecting factors such as Portability, Reliability, Complexity, Efficiency, Quality, Security, Cost, Maintainability, Cohesion, Availability and Flexibility along with their related attribute metrics. This paper has composed with four major objectives such as: the comparative analysis of Fuzzy Logic Control System and Neural Networks with their advantages and execution flow; The implications of Fuzzy Logic Control Neural Network Hybrid System architecture design for concern problem; The proposed FLCNNHS based algorithm and execution data flow diagram for executing the considered software reusability effecting factors along with their supporting attributes metrics for identification and Classification of Reusability Components through Strong, Week Relationships of Lattice Factors which is implacable for designing the better quality software product; and described the experimental analysis and results through proposed algorithmic approach. This innovative approach is more helpful for software developers to choose highly accurate components which are more required to build the high efficiency secure systems.

Keywords: *Fuzzy Logic Control System (FLCS), Fuzzy Logic Control Neural Network Hybrid System (FLCNNHS), Neural Network (NN), Prior-Knowledge (PK).*

1. Introduction

In fast growing life, the software developers always planning to use the software components which are straightforwardly entrenched into the past existing software applications and highly involve in the development of the high quality software product by reducing the production cost, time, user efforts, maintenance and man-power. The software reusability mode is highly required field for successful execution of artificial intelligence, machine learning based applications to fulfill the present and future human needs. In present scenario, the mainstreams of software systems are building by enhancing from past designed reusable components. The identification, classification and measuring the required components are key-roles concerns in fast development of software reusability components for producing the high quality software.

The main aspire of component implicated software development (CISD) is to decrease human-efforts, production-cost, expansion-time and which is used to produce the high superiority software invention [1]. The various excellence replicas are projected to forecast, develop the higher efficiency software applications. These quality models are implicated as a tool by holding main concern of the development of the new software applications. These entire implicated quality model are nonspecific models and which are not consider the quality aspects of CISD [10]. The reusability based components will be implicated based on the various quality effecting factors such as usability, reliability, flexibility, functionality, utility, configurability, compatibility, portability, understandability, cost, efficiency and so on.

The various authors are proposed the various methodologies in CISD environment by considering the various software component developments quality factors and their enhancement implications [15-33]. Recently the author Shrddha Sagar et. al. has proposed the approach of Software Quality Estimation of Component Based Software System by Using Fuzzy MOORA Approach by implicating with six indirectly or directly effecting superiority factors such as reliability, functionality, fault tolerance, security, efficiency and maintainability [10].

This paper is proposing the Fuzzy Logic Controller Neural Network Hybrid System which is implicated to recognize the affecting factors of component reusability execution by instituting the strong, week relationships in among these considered factors to fulfill the user requirement. This approach considered ten effecting factors such as usability, reliability, efficiency, compatibility, configurability, security, cost, portability, self-customizability and flexibility.

Researchers are proposed many algorithms and techniques have been developed for estimating the reliability of component-based software systems (CBSSs), much more research is needed [3]. Accurate estimation of the reliability of a CBSS is difficult because it depends on two factors: component reliability and glue code reliability [3]. The reliability is a real-world phenomenon with many associated real-time problems. Soft computing techniques can help to solve problems whose solutions are uncertain or unpredictable [3]. A number of soft computing approaches for estimating CBSS reliability have been proposed. These techniques learn from the past and capture existing patterns in data. The two basic elements of soft computing are neural networks and fuzzy logic [3]. The CISD environment is the process of assembling existing software components in an application such that they satisfy a predefined functionality. To assess the reuse of component, it is important to estimate reusability of these components [2].

2. Paper Objectives

This paper is proposing the Fuzzy Logic Controller Neural Network Hybrid System which is implicated to recognize the affecting factors of component reusability execution by instituting the strong, week relationships in among these considered factors to fulfill the user requirement. This is proposing four major objectives such as

- The comparative analysis of Fuzzy Logic Control System and Neural Networks with their advantages and execution flow;
- Implications of Fuzzy Logic Control Neural Network Hybrid System (FLCNNHS) architecture design;
- Proposed FLCNNHS based algorithm and execution data flow diagram for executing the considered software reusability effecting factors along with their supporting attributes metrics for identification and Classification of Reusability Components through Strong, Week Relationships of Lattice Factors which is implacable for designing the better quality software product.
- Experimental analysis and results through the proposed algorithmic approach.

3. Implications of Fuzzy Logic Control Neural Network Hybrid System (FLCNNHS)

The Soft Computing is an approach for building the systems which are computationally intelligent acts like as human brain by holding the special features of self-thinking, self-sensing, self-decision making and self-molding properties for solvating the complex problems. The hybridized intellectual experts systems are formed with a various range of artificial intelligence (AI) based neural network (NN) features by using the enhanced versions of fuzzy logic controller (FLC) system and genetic algorithm (GA). These two individual methods are represented by their own techniques, functionalities, objectives and methodology implications for solving the current complex problems [34]. By avoiding the individual limitations, the AI based NN is deriving the best future of function approximation along with their learning-training capabilities and the FLC is very good at decision making approach along with knowledge representation based computations. The neural based fuzzy logic system is used to impersonate the human actions directed with their ideas [34].

The contribution of metrics to the overall objective of the software quality is understood and recognized [4-7]. But how these metrics collectively determine reusability of a software component is still at its naïve stage [4]. A neural Network approach could serve as an economical, automatic tool to generate reusability ranking of software [8]. But, when one designs with Neural Networks alone, the network is a black box that needs to be defined, which is a highly compute-intensive process. One must develop a good sense, after extensive experimentation and practice, of the complexity of the network and the learning algorithm to be used. Fuzzy systems, on the other hand, require a thorough understanding of the fuzzy variables and membership functions, of the input-output relationships, as well as the good judgment to select the fuzzy rules that contribute the most to the solution of the application. As for the Fuzzy inference system there is a need of membership rules for fuzzy categories [4]. It is difficult to deduce these membership rules with a given set of complex data. Neural nets and fuzzy systems, although very different, have close relationship: they work with impression in a space that is not defined by crisp, deterministic boundaries [9]. Neural network can be used to define fuzzy rules for the fuzzy inference system [4]. A neural network is good at discovering relationships and pattern in the data, so neural network can be used to preprocess data in the fuzzy system. Furthermore, neural network that can learn new relationships with new input data can be used to refine fuzzy rules to create fuzzy adaptive system. With the objective of taking advantage of the features of the both, we proposed Neuro-Fuzzy based approach to identify reusable components in existing systems [4].

4. Proposed Methodology of Fuzzy Logic Control Neural Network Hybrid System (FLCNNHS)

The derived combinational mixed-version of neural fuzzy logic control system with inference rules composes the technical derivations to solve critical problems mixed connective conjunct consonants and mixed numerals. NFLCSIR touch all deviations of logical, numeric and linguistic data values. Knowledge of expert is easily adjustable by using the neural networks and with generation of additional rules of fuzzy membership functions is used to fulfill the certain specifications. It gives reasonable, flexible outputs and saves the time consumption period. As shown in the fig. 3. The imprecise input values can be converted into neuron based inputs such as A(i1-in), then these inputs transformed into fuzzy input to form the fuzzy sets [31]. Then the fuzzification approach has implemented with fuzzy based inference rules M(A) for prior collected hug fuzzy set formed of data sets A(i1-in) based on the fuzzy membership functions and fuzzy approached rules and it derives the values of M(B). It generates the linguistic values such as B1-Bn based on fuzzy processed inference-rule based fuzzy set. Then derive and compare the output responses of linguistic values with required input-linguistic values. The output responses (B) can be formed when the linguistic values satisfies the required input-linguistic values. If not, When the sensor detects any unrecognition of output or failure to recognize then the normalization process can be applied to output linguistic values to reduce the redundancy errors. The above mention process can be repeated when the sensor detects any overlapping recognition or slot correction errors.

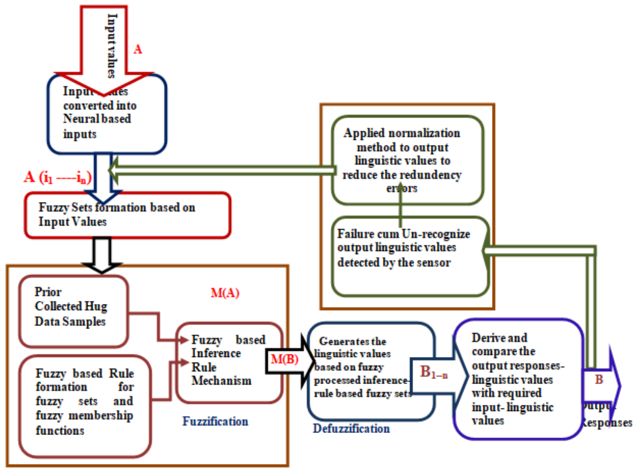

Fig. 1: Architecture of Fuzzy Logic Control Neural Network Hybrid System (FLCNNHS) with Inference Rules.

The FLCNNHS is trained with driven data learning method derived from the theory of neural network. In FLCNNHS, the fuzzy logic system determines the parameters by processing data samples by using learning algorithm generated from the theory of ANN. The set of fuzzy rules can represent at every stage of leaning process when it is before, during or after. Fuzzy rules can be interpreted as imprecise prototypes of training data. The underlying fuzzy system and leaning procedure in build for ensuring the semantic properties [14] as shown in the figure 2 and figure 3.
According to J-S. R. Jang [13], A fuzzy system can be considered to be a parameterized nonlinear map, called f, which can be expressed as below [4]

$$f(x) = \frac{\sum_{l=1}^{m} y^l \left(\prod_{i=1}^{n} \mu_{A_i^l}(x_i) \right)}{\sum_{l=1}^{m} \left(\prod_{i=1}^{n} \mu_{A_i^l}(x_i) \right)}$$

where yl is a place of output singleton, if Mamdani reasoning is applied or a constant, if Sugeno reasoning is applied. The membership function µAil(xi) corresponds to the input x=[x1, x2, x3,… xm] of the rule l . The "and" connective in the premise is carried out by a product and defuzzification by the center-of-gravity method. Consider a Sugeno type of fuzzy system having the rule base [4].

Rule1: If x is A1 and y is B1, then f1= p1x + q1y + 1
Rule2: If x is A2 and y is B2, then f2= p2x+ q2y + r2

Let the membership functions of fuzzy sets Ai, Bi, i=1,2, be ,µAi , µBi .

-Evaluating the rule premises results in wi =µAi(x) * µBi (y) where i = 1,2 for the rule rules stated above.
-Evaluating the implication and the rule consequences gives

$$f = \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2}$$

$$\overline{w_i} = \frac{w_i}{w_1 + w_2}$$

Then f can be written as

$$f = \overline{w_1} f_1 + \overline{w_2} f_2$$

In the FLCNNHS using a given input-output data set, we have constructed a fuzzy inference system (FIS) whose membership function parameters are tuned (adjusted) using stochastic gradient descent rule with momentum for the parameters associated with the input membership functions [4]. The initial rule-base for the neuro-fuzzy system can be obtained using of the ID3 decision tree Generation algorithm. As a result, the training error decreases, at least locally, throughout the learning process [4].

5. Experimental Analysis

In the current research a new approach to predict component reusability. The Fuzzy Inference System (FIS) and Artificial Neural Network (ANN) have already been used by many researchers using different factors of reusability every time. It is proven in many proposed techniques that neuro-fuzzy give the better results as compare to standalone FIS or ANN because it uses the power of rules decision of FIS and adaptive nature of ANN in a single system together [11]. This paper is proposing the Fuzzy Logic Controller Neural Network Hybrid System which is implicated to recognize the affecting factors of component reusability execution by instituting the strong, week relationships in among these considered factors to fulfil the user requirement.

5.1. Considered Related Effecting Factors

This approach considered eleven effecting factors such as Portability, Reliability, Complexity, Efficiency, Quality, Security, Cost, Maintainability, Cohesion, Availability and Flexibility along with their related attribute metrics. All supporting software reusability effecting factors are represented as row-wise execution and all supporting related effecting factors implicated attribute metrics are represented as column-wise execution as shown in the table 2 and figure 1.

5.2. Associated Effecting Factors Implicated Attribute Metrics

All title and author details must be in single-column format and must be centred.

Portability: Platform Independent, Machine Independent
Reliability: High Performance, Less Time
Complexity: Easy Understandable, Easy Adaptable
Efficiency: Minimum Resources, Fast Execution
Quality: Error-Free, Bug-Free, Test Cases, High Performance
Security: High Defence, High Privacy
Cost: low price
Maintainability: Easy Adaptable, Self- Mould
Cohesion: Module Bound
Availability: Easy Retrieve, Fast Retrieve
Flexibility: Machine Independent, Platform Independent, Self-Mould, Easy Adaptable

In this approach, all effecting factors are considered as input parameters which is converted into neural inputs and transmitted to the fuzzy logic controller which is used generates the linguistic values based on the neural inputs, the fuzzification inference rules implicated to derive the strong, week relationships then transmitted it to defuzzification stage to generate the outputs of accurate reusable components, then transmitted it final stage of neural mode to generated the output responses as shown in the algorithm and fig.4.

Table 1: All supporting software reusability effecting factors are represented as row-wise execution and all supporting related effecting factors implicated attribute metrics are represented as column-wise execution

	Platform Independent (PI)	Machine Independent (MI)	High Performance (HP)	Less Time (LT)	Easy Understandable (EU)	Easy Adaptable (EA)	Low Price (LP)	Self-Mould (SM)	Min-Resources (MR)	Fast Execution (FE)	Module Bound (MB)	Fast Retrieve (FR)	Error-Free (EF)	Bug-Free (BuF)	Test Cases (TC)	High Defence (HD)	High Privacy (HP)	Easy Retrieve (ER)
Portability (Port)	1	0.9	0.75	0.4	0.8	0.85	0.7	0.65	0.62	0.58	0.6	0.5	0.1	0.2	0.3	0.45	0.34	0.52
Reliability (Reli)	0.9	0.65	1	0.95	0.6	0.5	0.45	0.4	0.3	0.85	0.2	0.1	0.82	0.8	0.75	0.73	0.7	0.55
Availability (Avai)	0.35	0.3	0.25	0.85	0.2	0.15	0.1	0.8	0.78	0.12	0.75	0.9	0.7	0.65	0.55	0.45	0.4	1
Cohesion (Cohe)	0.65	0.6	0.55	0.2	0.72	0.7	0.1	0.5	0.9	0.95	1	0.45	0.85	0.8	0.75	0.4	0.35	0.3
Efficiency (Effi)	0.45	0.4	0.65	0.75	0.6	0.55	0.35	0.5	1	0.9	0.7	0.65	0.88	0.85	0.8	0.3	0.2	0.1
Maintainability (Main)	0.8	0.75	0.3	0.25	0.85	1	0.1	0.95	0.9	0.2	0.7	0.65	0.45	0.4	0.35	0.55	0.5	0.6
Security (Secu)	0.9	0.85	0.8	0.5	0.4	0.75	0.35	0.7	0.45	0.25	0.65	0.3	0.6	0.55	0.1	1	0.95	0.2
Cost	0.9	0.85	0.4	0.35	0.45	0.5	1	0.55	0.95	0.3	0.6	0.2	0.7	0.65	0.15	0.8	0.75	0.1
Quality (Qual)	0.6	0.55	0.8	0.5	0.2	0.3	0.25	0.25	0.75	0.7	0.65	0.35	1	0.9	0.85	0.45	0.4	0.1
Flexibility (Flex)	0.95	1	0.75	0.7	0.8	0.85	0.65	0.9	0.6	0.55	0.35	0.3	0.25	0.2	0.1	0.5	0.45	0.4
Complexity (Comp)	0.75	0.7	0.3	0.2	1	0.9	0.25	0.85	0.55	0.35	0.4	0.8	0.5	0.45	0.15	0.65	0.6	0.1

	Platform Independent (PI)	Machine Independent (MI)	High Performance (HP)	Less Time (LT)	Easy Understandable (EU)	Easy Adaptable (EA)	Low Price (LP)	Self-Mould (SM)	Min-Resources (MR)	Fast Execution (FE)	Module Bound (MB)	Fast Retrieve (FR)	Error-Free (EF)	Bug-Free (BuF)	Test Cases (TC)	High Defence (HD)	High Privacy (HP)	Easy Retrieve (ER)
Portability (Port)	1	0.9	0.75	0.4	0.8	0.85	0.7	0.65	0.62	0.58	0.6	0.5	0.1	0.2	0.3	0.45	0.34	0.52
Reliability (Reli)	0.9	0.65	1	0.95	0.6	0.5	0.45	0.4	0.3	0.85	0.2	0.1	0.82	0.8	0.75	0.73	0.7	0.55
Availability (Avai)	0.35	0.3	0.25	0.85	0.2	0.15	0.1	0.8	0.78	0.12	0.75	0.9	0.7	0.65	0.55	0.45	0.4	1
Cohesion (Cohe)	0.65	0.6	0.55	0.2	0.72	0.7	0.1	0.5	0.9	0.95	1	0.45	0.85	0.8	0.75	0.4	0.35	0.3
Efficiency (Effi)	0.45	0.4	0.65	0.75	0.6	0.55	0.35	0.5	1	0.9	0.7	0.65	0.88	0.85	0.8	0.3	0.2	0.1
Maintainability (Main)	0.8	0.75	0.3	0.25	0.85	1	0.1	0.95	0.9	0.2	0.7	0.65	0.45	0.4	0.35	0.55	0.5	0.6
Security (Secu)	0.9	0.85	0.8	0.5	0.4	0.75	0.35	0.7	0.45	0.25	0.65	0.3	0.6	0.55	0.1	1	0.95	0.2
Cost	0.9	0.85	0.4	0.35	0.45	0.5	1	0.55	0.95	0.3	0.6	0.2	0.7	0.65	0.15	0.8	0.75	0.1
Quality (Qual)	0.6	0.55	0.8	0.5	0.2	0.3	0.25	0.25	0.75	0.7	0.65	0.35	1	0.9	0.85	0.45	0.4	0.1
Flexibility (Flex)	0.95	1	0.75	0.7	0.8	0.85	0.65	0.9	0.6	0.55	0.35	0.3	0.25	0.2	0.1	0.5	0.45	0.4
Complexity (Comp)	0.75	0.7	0.3	0.2	1	0.9	0.25	0.85	0.55	0.35	0.4	0.8	0.5	0.45	0.15	0.65	0.6	0.1

Fig. 2: All supporting software reusability effecting factors are represented as row-wise execution and all supporting related effecting factors implicated attribute metrics are represented as column-wise execution

In table 2 and fig 4 the yellow background coloured numbers as considered as the strong relationship metrics of attributes of concern software reusability factors.

5.3. FLCNNHS Methodology Execution Flow

In the proposed model for reusability, the input variables which have been taken, each consisting of three different fuzzy linguistic sets such as high, medium and by using Gaussian- membership-functions such as μ Z(xk, yk) along with Fuzzy controller in a closed-loop configuration implications. The fuzzification process is applied on these sets by taking the normalized values for concern attributes by implicating membership values 0 and 1 and between [0 1] and for defuzzification, so as to produce a crisp value, five membership functions range is defined between [0 1] such as very low, low, medium high, very high for reusability index. In this approach, all effecting factors are considered as input parameters which is converted into neural inputs and transmitted to the fuzzy logic controller which is used generates the linguistic values based on the neural inputs, the fuzzification inference rules implicated to derive the strong, week relationships then transmitted it to defuzzification stage to generate the outputs of accurate reusable components, then transmitted it final stage of neural mode to generated the output responses. The given FLCNNHS system based on expert knowledge contains 14 rules, 6 inputs and one single output for reusability.

The entire data set of reusability level is 27 samples. They are referred to as training data and testing data. Upon training, the ANFIS shows the training error which reflects that how good the mapping function is working. To validate the model, further applied the testing data to see that how the ANFIS behaves for known data. ANFIS maps the function onto the testing data as per the training. This means a new FIS is created to fit the data into membership functions using the grid partitioning method as it generates a single-output Sugeno-type FIS. The ANFIS automatically selects the membership function and also generates the new FIS [11]

5.4. Algorithmic Design

Step1: Considered Effecting Reusability Factors

Step2: Case Study1: Recognize the applicable Effecting Reusability Factors

Step3: Case Study2: Recognize the related attribute metrics of considered applicable Effecting Reusability Factors from case study1

Step4: Classification: Classify the metrics based on required objectives with help of Neural Networks Mechanism

Step5: Conversion of input attribute metrics values into neural inputs for easy classification

Step6: Conversion of neural inputs into fuzzy input representation

Step7: Define the input parameters, membership functions

Step8: Generate the Fuzzy Inference Rules

Step9: Measuring the parameters
First Strong Relationship Metric →SRM1
Second Strong Relationship Metric →SRM2
First Week Relationship Metric →WRM1
Second Week Relationship Metric →WRM2

Step10: Define the Reusability Item Set and Non-Reusability Item Set
Reusability Item Set →RIS
Non-Reusability Item Set →NRIS

Step11: Based on Fuzzy Logic Inference Rules the If-Then Implications are applied

Step12: The efficient execution: either RIS based Execution (Step13) or NRIS based Execution (Step14)

Step13: The RIS based Execution
If RIS = f (SRM1, SRM2) Then
Then Best Reusability Item Set (BRIS) ←RIS
To process BRIS for software reusability
Otherwise
Generate the different training, learning test cases

Step14: The NRIS based Execution
If NRIS = f (WRM1, WRM2) Then
Then Best Avoidable Reusability Item Set (BARIS) ←NRIS
To process BARIS for software reusability
Otherwise
Generate the different training, learning test cases

Step15: Choose the best parameters of attribute metrics values based on either BRIS implication or BARIS implication.

5.4. Data Flow Execution

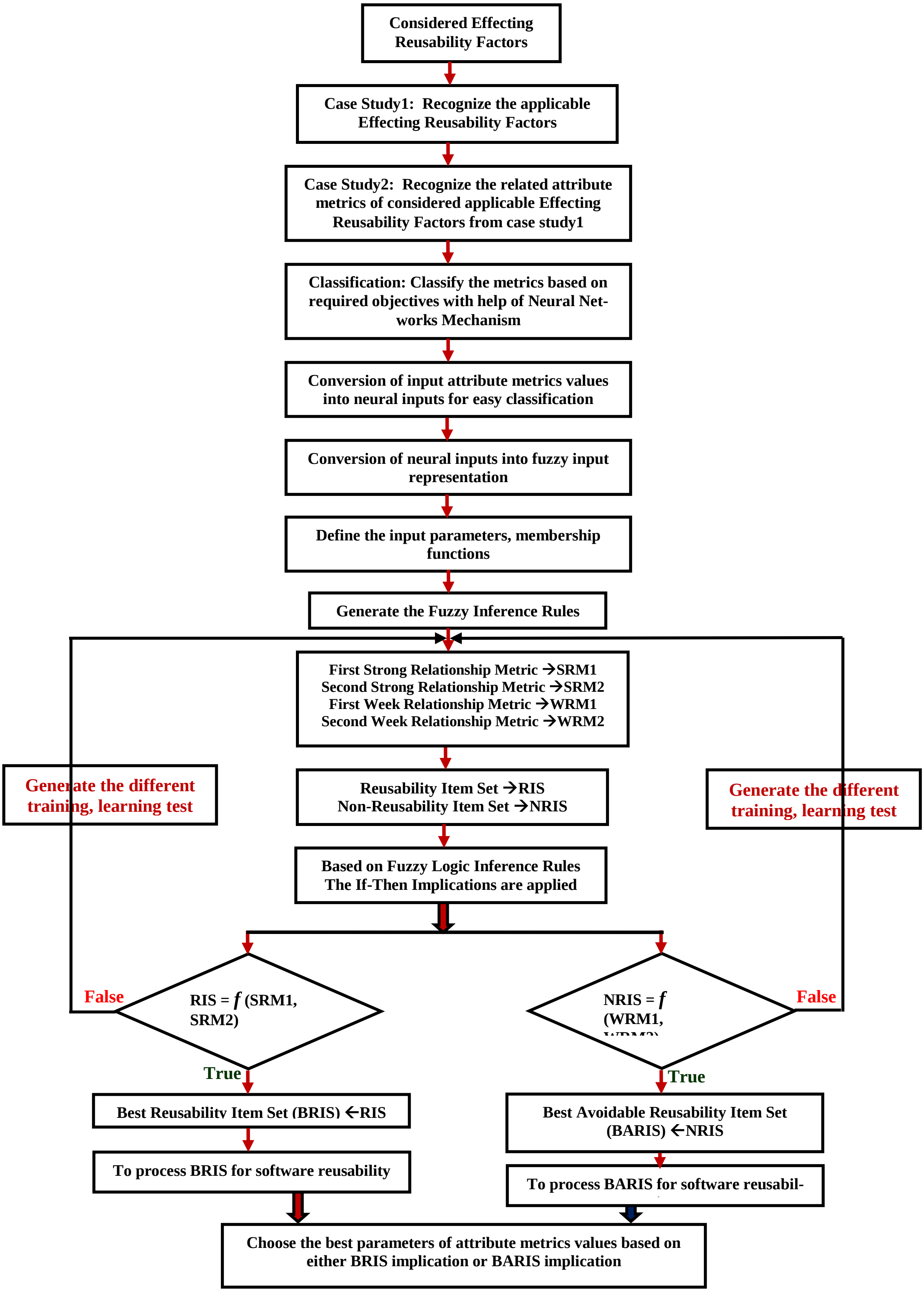

Fig. 3: Data Flow Execution of Identification, Classification of Reusability Components through Strong, Week Relationships of Lattice Factors

5.6 Experimental Results

	PI	MI	HP	LT	EU	EA	LP	SM	MR	FE	MB	FR	EF	BF	TC	HD	HP	ER
Port	1	0.9	0.75	0.4	0.8	0.85	0.7	0.65	0.2	0.58	0.6	0.5	0.1	0.2	0.3	0.45	0.34	0.52
Reli	0.9	0.65	1	0.95	0.6	0.5	0.45	0.4	0.3	0.85	0.2	0.1	0.82	0.8	0.75	0.73	0.7	0.55
Avai	0.35	0.3	0.25	0.85	0.2	0.15	0.1	0.8	0.78	0.12	0.75	0.9	0.7	0.65	0.55	0.45	0.4	0.1
Cohe	0.65	0.6	0.55	0.2	0.72	0.7	0.1	0.5	0.9	0.95	1	0.45	0.85	0.8	0.75	0.4	0.35	0.3
Effi	0.45	0.4	0.65	0.75	0.6	0.55	0.35	0.5	1	0.9	0.7	0.65	0.88	0.85	0.8	0.3	0.2	0.1
Main	0.8	0.75	0.3	0.25	0.85	1	0.1	0.95	0.9	0.2	0.7	0.65	0.45	0.4	0.35	0.55	0.5	0.6
Secu	0.9	0.85	0.8	0.5	0.4	0.75	0.35	0.7	0.45	0.25	0.65	0.3	0.6	0.55	0.1	1	0.95	0.2
Cost	0.9	0.85	0.4	0.35	0.45	0.5	1	0.55	0.95	0.3	0.6	0.2	0.7	0.65	0.15	0.8	0.75	0.1
Qual	0.6	0.55	0.8	0.5	0.2	0.3	0.25	0.22	0.75	0.7	0.65	0.35	1	0.9	0.85	0.45	0.4	0.1
Flex	0.95	1	0.75	0.7	0.8	0.85	0.65	0.9	0.6	0.55	0.35	0.3	0.25	0.2	0.1	0.5	0.45	0.4
Comp	0.75	0.7	0.3	0.2	1	0.9	0.25	0.85	0.55	0.35	0.4	0.8	0.5	0.45	0.15	0.65	0.65	0.1

Fig. 4: Experimental table of all supporting software reusability effecting factors and its supporting related effecting factors implicated attribute metrics as shown in the table 2 and fig.3.

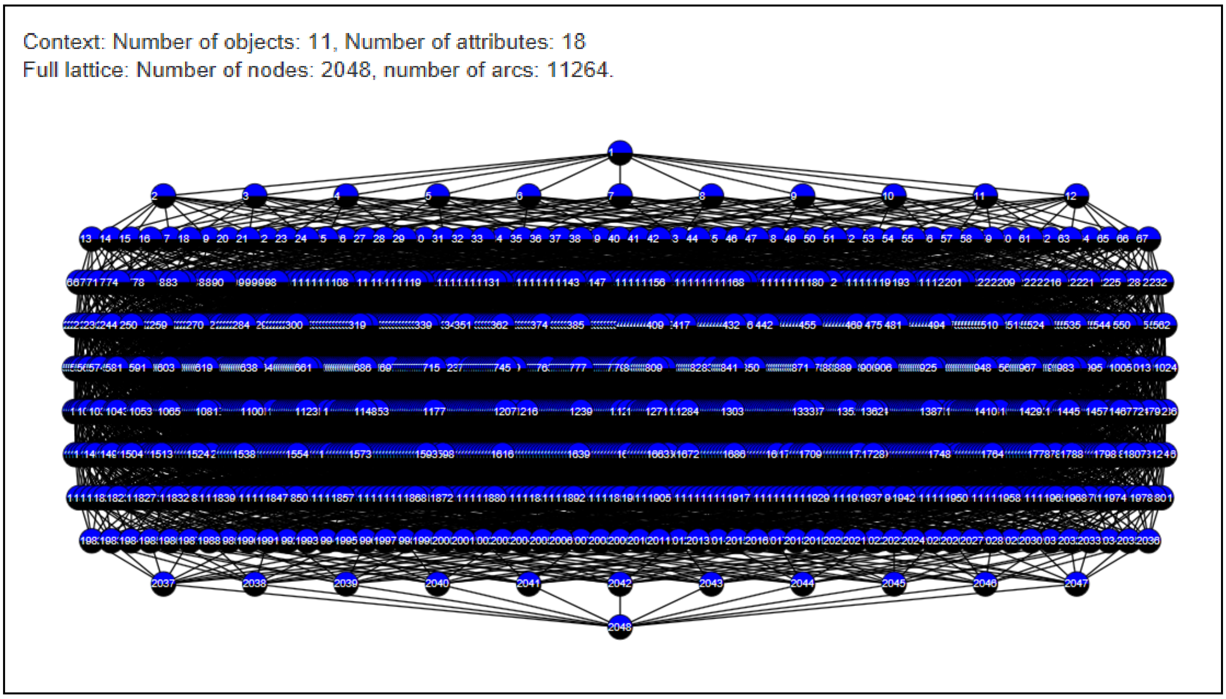

Fig. 5: Experimental scenario executed from fig.5 to generate the relationships through the considered objects of software reusability effecting factors and it supporting attributes metrics.

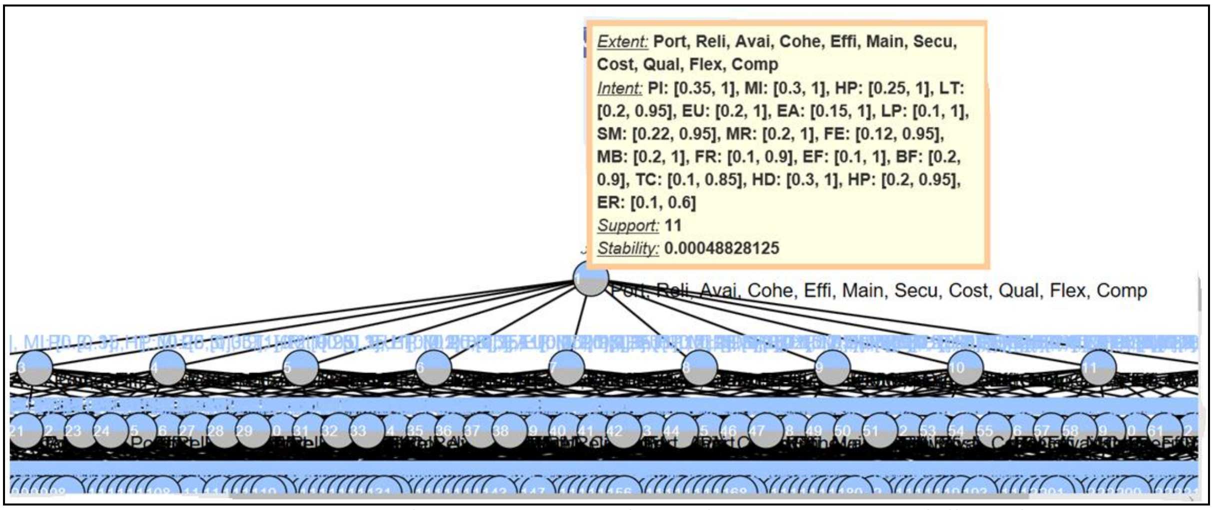

Fig. 6: Experimental scenario of node to node execution for identification overall stability of effecting factors.

The figure 5 is representing the Experimental scenario executed from fig.5 to generate the relationships through the considered objects of software reusability effecting factors and it supporting attributes metrics. It has generated the relational lattices of objects of software reusability effecting factors to it supporting attributes metrics through 2048 nodes and 11264 supporting arcs. The accurate relationships are identified through the particular objective based node to node relationship as shown in the fig.6.

6. Conclusion

In present scenario, the mainstreams of software systems are building by enhancing from past designed reusable components. The identification, classification and measuring the required components are key-roles concerns in fast development of software reusability components for producing the high quality software. This paper is proposing the Fuzzy Logic Controller Neural Network Hybrid System which is implicated to recognize the affecting factors of component reusability execution by instituting the strong, week relationships in among these considered factors to fulfil the user requirement. This approach considered eleven effecting factors such as Portability, Reliability, Complexity, Efficiency, Quality, Security, Cost, Maintainability, Cohesion, Availability and Flexibility along with their related attribute metrics. This paper has implemented with four major objectives such as: the comparative analysis of Fuzzy Logic Control System and Neural Networks with their advantages and execution flow; The implications of Fuzzy Logic Control Neural Network Hybrid System (FLCNNHS) architecture design for concern problem; The proposed FLCNNHS based algorithm and execution data flow diagram for executing the considered software reusability effecting factors along with their supporting attributes metrics for identification and Classification of Reusability Components through Strong, Week Relationships of Lattice Factors which is implacable for designing the better quality software product; and described the experimental analysis and results through proposed algorithmic approach. This innovative approach is more helpful for software developers to choose highly accurate components which are more required to build the high efficiency secure systems.

Acknowledgement

I would like to express my gratitude to my guide Dr B.Rama for her kind hearted support, help and cooperation.

References

[1] Arun Sharma, P. S. Grover, Rajesh Bhatia, Reusability Assessment for Software Components – a Neural Network Based Approach, ACM SIGSOFT Software Engineering Notes 34(2):1-6, DOI: 10.1145/1507195.1507215, February 2009

[2] Shrddha Sagar, N. W. Nerurkar, Arun Sharma, “A soft computing based approach to estimate reusability of software components”, ACM SIGSOFT Software Engineering, ACM New York, NY, USA, doi:10.1145/1811226.1811235, Volume 35 Issue 4, Pages 1-4, July 2010.

[3] Kirti Tyagi, Arun Sharma, “An adaptive neuro fuzzy model for estimating the reliability of component-based software systems”, Applied Computing and Informatics, Production and hosting by Elsevier, pp:38–51, (2014) 10,

[4] Parvinder S. Sandhu and Hardeep Singh, “Automatic Reusability Appraisal of Software Components using Neuro-fuzzy Approach”, World Academy of Science, Engineering and Technology International Journal of Computer and Systems Engineering, Vol:1, No:8, 2007.

[5] W. Humphrey, Managing the Software Process, SEI Series in Software Engineering, Addison-Wesley, 1989.

[6] L. Sommerville, Software Engineering, Addision-Wesley, 4th Edition, 1992.

[7] R. S. Pressman, Software Engineering: A Practitioner’s Approach, McGraw-Hill Publications, 5th edition, 2005.

[8] G. Boetticher and D. Eichmann, “A Neural Network Paradigm for Characterising Reusable Software,” Proceedings of the 1st Australian Conference on Software Metrics, 18-19 November 1993.

[9] S. V. Kartalopoulos, Understanding Neural Networks and Fuzzy Logic-Basic Concepts and Applications, IEEE Press, 1996, pp. 153-160.

[10] Shrddha Sagar, Pratistha Mathur and Arun Sharma, “Software Quality Estimation of Component Based Software System by Using Fuzzy MOORA Approach”, International Journal of Artificial Intelligence and Application for Smart Devices, ISSN: 2288-6710 IJAIASDVol.5, No.1 (2017), pp.21-30, 2017.

[11] Shalini Goel, Arun Sharma, Neuro Fuzzy based Approach to Predict Component’s Reusability, International Journal of Computer Applications (0975 – 8887) Volume 106 – No.5, November 2014

[12] SOFTWARE: PRACTICE AND EXPERIENCE Softw. Pract. Exper. 2017; 47:941–942, Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/spe.2504, Published online 12 May 2017.

[13] J-S. R. Jang and C.T. Sun, “Neuro-fuzzy Modeling and Control,” Proceeding of the IEEE, March 1995.

[14] Santosh Kumar Henge, Dr B.Rama, “Neural Fuzzy Closed Loop Hybrid System for Classification, Identification of Mixed Connective Consonants and Symbols with Layered Methodology” ((ICPEICES-2016)”, Pages: 2880-2887 and published in IEEE Xplore: DOI: 10.1109/ICPEICES.2016.7853708, INSPEC accession number: 16672805, ISBN: Electronic ISBN: 978-1-4673-8587-9, Print on Demand (PoD) ISBN: 978-1-4673-8588-6, added date: 16 Feb 2017.

[15] Nagib C. Callaos B. (1994) ―Designing with Systemic Total Quality‖ Educational Technology Vol. 34, Issue.1, Page(s) 29-36.

[16] Grady R B. (1992) ―Practical Software Metrics for Project Management and Process improvement Publication: Book , Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1992.

[17] Tomar P. and Gill N. ― New Algorithm for Component Selection to Develop Component-Based Software with X Model‖ Lecture Notes on Software Engineering COTS Component Selection‖ Journal of Object Technology, Vol. 4, 2005.

[18] Alvaro A. de Almeida E.S. ; Meira S.L. ―A Software Component Quality Model: A Preliminary Evaluation‖ Published in: Software Engineering and Advanced Applications. SEAA '06. 32nd EUROMICRO Conference, Page(s): 28-37, Cavtat, Dubrovnik, 2006.

[19] Kumar, Vijai, Arun Sharma, Rajesh Kumar, and P. S. Grover, Quality aspects for component- based systems: A metrics based approach Software: Practice and Experience, pp. 1531-1548, 2012.

[20] McCall,J. A., Richards P K. and Walters G F. ―Factors in Software Quality‖ Springfield VA National Technical Information Service, Vol. 3, Issue. 5, Page(s): 133-139, 1977.

[21] Sharma, A., Kumar, R., Grover, P. S., Empirical Evaluation and Validation of Interface Complexity Metrics for Student Components, International Journal of Software Engineering and Knowledge Engineering, Vol. 18, Issue 7, pp. 919—931, 2008.

[22] Boehm B W. ―Characteristics of Software Quality TRW Series of software Technology‖ Amsterdam North Holland, 1978

[23] M.Bertoa and A. Vallecillo ―Quality Attributes for COTS Components‖ proceedings of the 6th International ECOOP Worshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE), Malaga, 2002.

[24] Pande J. ―On Some Critical ssues in Component Selection in Component based Software Development‖ nternational Journal of Computer Applications, Vol. 46, pp. 0975-8887, 2012.

[25] Pande.J., Bisht.R.K., Pant.D. Pathak.V.K. ―On Some Quality ssues of Component Selection in CBSD‖ J. Software Engineering & Applications Vol. 3 pp. 556-560, 2010.

[26] Dromey R.G. (1995) ―A Model for Software Product Quality‖ Published in: EEE Transactions on Software Engineering, Vol. 21 , Issue. 2, Page(s): 146 – 162.

[27] Ince D. ― ISO 9001 and Software Quality Assurance‖ McGraw-Hill, New York, 1994.

[28] Bertoa M F. and Vallecillo A. ―Usability Metrics for Software Components‖ QAOOSE’04:Proceedings of the 8th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering. Page(s): 1- 10, 2004.

[29] Kumar R. Sharma A. and Grover P.S. ―Predicting Maintainability of Component-based Systems by using Fuzzy-Logic‖ Communications in Computer and information Science Springer Berlin Heidelberg, USA, Vol. 40, Issue 11, pp. 581-593, 2009.

[30] Lloyd, Wesley James, "A Common Criteria Based Approach for COTS Component Selection", Journal of Object Technology, V0l. 4, pp. 25-32, 2005.

[31] Santosh Kumar Henge, Dr B.Rama, “Five Layered-Neural Fuzzy Closed Loop Hybrid Control System with Compound Bayesian Decision Making Process for Classification cum Identification of Mixed Connective Conjunct Consonants and Numerals”, Published in “© Springer Nature Singapore Pte Ltd”. 2017, Advances in Computer and Computational Sciences, “Advances in Intelligent Systems and Computing-553”, pp.619-629, DOI: 10.1007/978-981-10-3770-2_58, 2017.

[32] Tyagi k. Sharma A. ―Significant Factors for Reliability Estimation of Component Based Software Systems‖ Journal of Software Engineering and Applications, Vol. 7, pp.934-942, 2014.

[33] Madaan N. and Kaur J. ― A Survey on Selection Techniques of Component Based Software‖ International Journal of Informatics and Computation technology, Vol. 4, pp. 1245-1250, 2014.

[34] Santosh Kumar Henge, Dr B.Rama, “Comparative Analysis of Soft Computing Hybrid Intellectual System Implications for Salvation of Machine Learning based Complex Problems”, in International Journal of Advanced Scientific Technologies, Engineering and Management Sciences (IJASTEMS-ISSN: 2454-356X) Volume.3,Special Issue.1, Page No: 15-27, March.2017.