

# Type-2 Fuzzy Logic Approach To Increase The Accuracy Of Software Development Effort Estimation

Zahra Barati<sup>1</sup>, Mehdi Jafari Shahbarzade<sup>2</sup>, Vahid Khatibi Bardsiri<sup>3</sup>

Received (2016-04-28)

Accepted (2016-11-16)

---

**Abstract** — predicting the effort of a successful project has been a major problem for software engineers the significance of which has led to extensive investigation in this area. One of the main objectives of software engineering society is the development of useful models to predict the costs of software product development. The absence of these activities before starting the project will lead to various problems. Researchers focus their attention on determining techniques with the highest effort prediction accuracy or on suggesting new combinatory techniques for providing better estimates. Despite providing various methods for the estimation of effort in software projects, compatibility and accuracy of the existing methods is not yet satisfactory. In this article, a new method has been presented in order to increase the accuracy of effort estimation. This model is based on the type-2 fuzzy logic in which the gradient descend algorithm and the neuro-fuzzy-genetic hybrid approach have been used in order to teach the type-2 fuzzy system. In order to evaluate the proposed algorithm, three databases have been used. The results of the proposed model have been compared with neuro-fuzzy and type-1 fuzzy system. This comparison reveals that the results of the proposed model have been more favorable than those of the other two models.

**Index Terms** — Fuzzy logic, Gradient descent, Neuro-Fuzzy, Software effort estimation, Type-2 fuzzy logic.

## I. INTRODUCTION

One of the main objectives of the software engineering society is smart model development in order to develop software life cycle and to predict the costs of developing software products [1][2][3].

Developing effort prediction models of software projects has led to purposeful studies in the past in order to estimate software development effort [2][4][5]. Generally, development effort estimation, as one of the three major challenges in computer sciences, is a major activity in software project planning [4][6]. Software development effort estimation can be considered as a subset of software estimates [7][2]. In the existing models of effort estimation, reducing relative error is considered as the most important goal, and it is attempted to reduce the amount of error as much as possible. Due to the uncertainty and sophisticated non-linear properties of software projects, high and reliable accuracy cannot be achieved by mere focusing on estimation model criterion. In addition, models based on single-objective optimization are not able to manage software projects, and the results of this type of estimators are greatly different from one database to another. As a result, due to the high level of complexity, it is impossible to generalize the accuracy of the existing effort estimators for various software projects [8][9]. Without a proper estimate of the required cost, the project manager cannot determine how much time, work force, and many other sources s/he needs in order to undertake the project. In case of misdiagnosis, the project will be doomed to certain failure. Surveys conducted indicate that most software

1- Msc Student, Computer Engineering Department, Kerman Branch, Islamic Azad University, Kerman, Iran. (zahrabarati68@yahoo.com).

2- Assistant Professor, Electrical Engineering Department, Kerman Branch, Islamic Azad University, Kerman, Iran.

3- Assistant Professor, Computer Engineering Department, Bardsir Branch, Islamic Azad University, Kerman, Iran.

projects have failed because of incorrect cost estimate as well as inadequate planning and timing. As a result, accurate estimation of software projects has taken on great significance [10][11][12]. Despite providing various methods for the estimation of effort in software projects, compatibility and accuracy of the existing methods is not yet satisfactory. Generally, the performance of the available approaches in different software projects is not the same and a wide range of performance discrepancies is evident. The present study provided a new approach increasing the estimation accuracy of software project effort. The proposed model is based on type-2 fuzzy logic applying two algorithms for training type-2 fuzzy system. As indicated by the literature, applying type-2 fuzzy logic can be helpful in effort estimation of software projects and increase the accuracy [1][7]. Section 2 of the present paper research instruments, section 3 presents the proposed model. In section 4, the evaluation parameters of the proposed model are introduced. Section 5 elaborates on the evaluation of the results, and conclusion is finally presented in section 6.

## II. RESEARCH INSTRUMENTS

### 1. Type-2 Fuzzy Logic

In 1975, Professor Lotfi Asgarzadeh introduced type-2 fuzzy collections as a developed version of fuzzy collections. Since type-2 fuzzy collections enjoy ranks of fuzzy membership, they are also called fuzzy-fuzzy collections capable of reducing uncertain effects and presenting a model in uncertainties. These collections are helpful in situations where the exact identification of membership ranking in fuzzy collections is assumed difficult. Figure 1 illustrates the structure of membership function in type-2 fuzzy logic [13][14][15] [16].

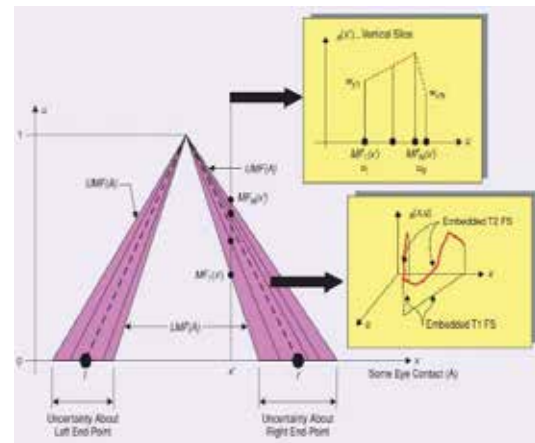


Fig 1: The structure of the membership function in type-2 fuzzy logic [10]

### 2. Neuro-Fuzzy

The neuro-fuzzy model, which is a combination of neural network and fuzzy logic, determines the parameters of the fuzzy system using the neural network learning algorithm. This combinatory system has been established based on the fuzzy system, indicative of uncertainty. Simultaneous neuro-fuzzy, cooperative neuro-fuzzy, and hybrid neuro-fuzzy are examples of types of neuro-fuzzy models. In hybrid neuro-fuzzy models, changes made in the learning process can be interpreted from both the perspective of the neural network and that of the fuzzy logic [17][18].

## III. PRESENTING THE PROPOSED MODEL

In this section, details of the proposed model is presented. In the proposed model, in order to teach the type-2 fuzzy logic, two methods have been used which will be discussed later in this section. The proposed model aims at determining the amount of effort required for every project using type-2 fuzzy logic. For this purpose, it is necessary to define the projects as inputs. When defining the inputs, it is necessary to determine the membership function of each input, and depending on the type of the membership function, the corresponding parameters must also be adjusted. In the proposed model, the type of the Gaussian membership function has been taken into consideration. In order to adjust these parameters, two different methods, gradient descent and neuro-fuzzy-genetic hybrid approach have been used as follows:

### 1. Presenting two algorithms for training of type-2 fuzzy system

#### III.1.1. Approach of descending gradient reduction

One of the approaches applied for training of type-2 fuzzy logic is descending gradient reduction. In this approach, first of all, modifiable parameters in type-2 fuzzy system are specified. In type-2 fuzzy system, modifiable parameters are related to input membership function. Based on their figure and relationship, membership functions have different modifiable parameters. In this approach, Adjustable membership function is applied. Adjustable parameters in this membership function are presented in Table 1.

TABLE 1  
ADJUSTABLE PARAMETERS IN GAVSI'S  
MEMBERSHIP FUNCTION

Center of Category/class	$\bar{x}_i$
Standard deviation of input membership function	$\sigma_i$
Membership rank	$ds$

#### III.1.2. Processing steps of the first suggested algorithm

1)  $\sigma_i, \bar{x}_i, ds$  Coefficients primary value-taking of Adjustable function (at the beginning process of algorithm, these coefficients are taken values randomly.). Also, learning rate and the maximum number of algorithm iteration (loop stop condition) are required to be determined in this step.

- 2) Start – training loops
  - inputting data into type-2 fuzzy system and giving the output
  - comparing output and goal value
  - estimating the level of error

3) Estimation of error assessment criterion

4) Controlling loop-stop condition

#### 2. The synthetic approach of neuro-fuzzy and genetic algorithm

The second approach applied for type-2 fuzzy system training is the synthetic approach of

neuro-fuzzy and genetic algorithm according to the following process:

Among all three parameters of Adjustable membership function, two parameters,  $\sigma_i$  &  $\bar{x}_i$  are determined by neuro-fuzzy algorithm while  $ds$  parameter is determined by genetic algorithm.

The processing steps of the second proposed approach is as follows:

- 1) Implementation of a neuro-fuzzy system on data.
- 2) Extracting mean and sigma properties of the final model of neuro-fuzzy.
- 3) Constructing a type-2 fuzzy system according to neuro-fuzzy output.
- 4) Implementing genetic algorithm on type-2 fuzzy system
- 5) Obtaining  $ds$  coefficient.
- 6) Final establishment of type-2 fuzzy system and obtaining output.

After adjusting the parameters related to the membership functions, the data are applied to the structure of type-2 fuzzy logic, and type-2 fuzzy system estimates the required effort for every project.

## IV. ASSESSMENT PARAMETERS IN THE PROPOSED APPROACH

The present study made use of the following three widely-used accuracy assessment criteria: The value of relative error indicating the difference between the estimated costs by algorithm and the real costs, mean of relative error shown by the exclusive abbreviation of MMER indicating the mean of estimation error for all study samples (either training or testing samples) and prediction percentage indicating the percentile of samples with estimation error of less or equal to X value.

$$MER_i = |\text{Actual-Estimation}| / \text{Estimation} \quad (1)$$

$$MMER = \sum_{i=1}^N MER_i / N \quad (2)$$

$$MdMRE = \text{Median} (MER_i) \quad (3)$$

$$\text{PRED}(X) = A/N \quad (4)$$

According to the above-mentioned statements, mean is estimated considering the value of each real effort and obtaining it out of total applied

data. The assessment of model strength can be deviated when there are several projects with big relative error. One alternative for mean is median. Median shows a centripetal assessment and is less sensitive to the existence of several big relative errors. In order to obtain median, data are listed in an ascending order. If the number of data are odd, the one in the middle of all is considered as median. If the number of data are even, the mean of the two data located at the middle of all is considered as the median. The estimation strength of effort estimation models is then measured by median.

In these formulas, A is the number of projects with the average relative error less than or equal to X, and N is the number of estimated projects. In software effort estimation method, acceptable level for X is 0.25, and the proposed methods are compared in accordance with this method.

Then, the results of the assessment criteria related to this model are compared with fuzzy and neuro-fuzzy methods. These results indicate that type-2 fuzzy logic has a better performance than the fuzzy and neuro-fuzzy methods. In Fig 2, the flowchart of the proposed model is presented.

## V. EXPERIMENTAL RESULTS

In this section, the results of the proposed model implementation through two proposed algorithms in three different data sets have been shown. The following tables show the results of the implementation of type-2 fuzzy logic model in three different data sets in comparison with other methods.

### 1. The evaluation of results on reference database [19]

According to Tables 2 and 3, the proposed model has managed to achieve a higher accuracy in all three assessment criteria in comparison with fuzzy and neuro-fuzzy methods.

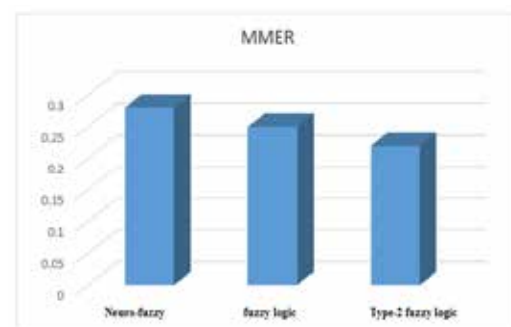
**TABLE 2:**  
**THE RESULTS OBTAINED ON THE REFERENCE DATABASE [19] IN THE GRADIENT DESCENT METHOD**

Methods	MMER	MdMRE	PRED
Neuro- fuzzy	0.28	0.24	0.5
Logic Fuzzy	0.25	0.31	0.41
Fuzzy Type_2	0.22	0.17	0.51

**TABLE 3:**  
**THE RESULTS OBTAINED ON THE REFERENCE DATABASE [19] IN THE NEURO-FUZZY-GENETIC HYBRID APPROACH**

Methods	MMER	MdMRE	PRED
Neuro-Fuzzy	0.61	0.36	0.32
Fuzzy Logic	0.45	0.79	0.10
Type_2 Fuzzy	0.33	0.27	0.43

Figure 3 illustrates the values of MMER parameters for the above-mentioned models. As indicated by figure 3, the best value (0.22) was obtained by type-2 fuzzy system and the worst value (0.28) was obtained by neuro-fuzzy model.



**Fig 3: Chart of different models' MMER results on reference database**

According to the obtained value for MdMRE parameter for different models in figure 4, the best value was obtained by type-2 fuzzy system model (0.17) and the worst one was obtained by fuzzy model (0.41). This parameter indicates relative error median.

Also, the best obtained value for PRED parameter is for fuzzy model (0.41). This parameter of the proposed model with the value of 0.21 is located at the 3rd rank. As indicated by the results of reference database, the proposed model performed better relative to other models.

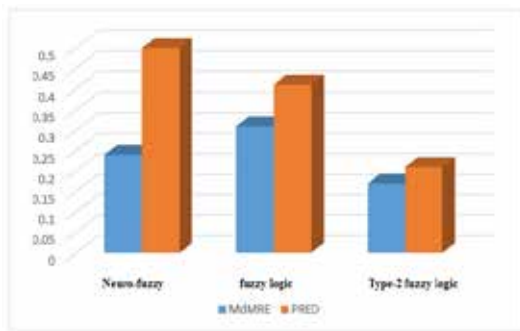


Fig4: Chart of different models’ MdMRE and PRED results on reference dataset

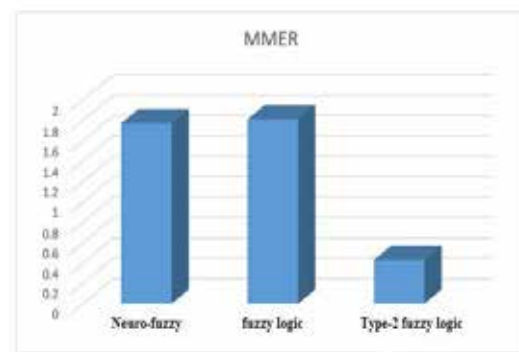


Fig 5: Char of different models’ MMER results on Albrecht’s dataset

2. Results on Albrecht’s database

Given the results presented in Table 4, the proposed model has managed to achieve the best results in all three parameters. In terms of MMER and PRED parameters, the lowest values related to the fuzzy logic were 1.81 and 0.006 respectively, and the lowest value of the MdMRE parameter related to the neuro-fuzzy model was 0.96.

TABLE 4:  
THE RESULTS OBTAINED ON ALBRECHT’S DATABASE IN THE GRADIENT DESCENT METHOD

Methods	MMER	MdMRE	PRED
Neuro-Fuzzy	1.77	0.96	0.20
Fuzzy Logic	1.81	0.78	0.006
Type 2 Fuzzy	0.43	0.91	0.25

TABLE 5:  
THE RESULTS OBTAINED ON ALBRECHT’S DATABASE IN THE NEURO-FUZZY-GENETIC HYBRID APPROACH

Methods	MMER	MdMRE	PRED
Neuro-Fuzzy	1.15	0.34	0.20
Fuzzy Logic	0.75	1.95	0.12
Type 2 Fuzzy	0.53	0.54	0.21

The results presented in Table 5 show that the proposed model ranked first in all three parameters, and in terms of the MMRE parameter, the lowest value belongs to the neuro-fuzzy model, and the lowest value for MdMR and PRED parameters belongs to fuzzy logic.

The obtained results on Albrecht’s dataset indicate that the proposed model performed better relative to other models. These results are illustrated in figures 5 and 6.

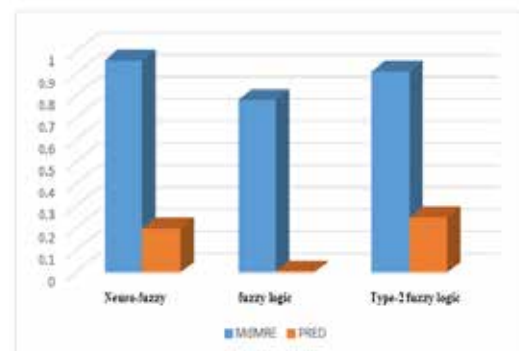


Fig 6: Char of different models’ MdMRE and PRED results on Albrecht’s dataset

3. Results on the Desharnais database

The results presented in Tables 6 and 7 show that the proposed model on the Desharnais dataset has the best performance compared to the other two methods while using the neuro-fuzzy-genetic hybrid approach.

TABLE 6:  
THE RESULTS OBTAINED ON THE DESHARNAIS DATABASE BY THE PROPOSED GRADIENT DESCENT ALGORITHM

Methods	MMER	MdMRE	PRED
Neuro-Fuzzy	0.41	1.73	0.24
Fuzzy Logic	0.42	1.50	0.21
Type 2 Fuzzy	0.27	0.33	0.45

TABLE 7:  
THE RESULTS OBTAINED ON THE DESHARNAIS IN THE NEURO-FUZZY-GENETIC HYBRID APPROACH

Methods	MMER	MdMRE	PRED
Neuro-Fuzzy	0.35	0.35	0.42
Fuzzy Logic	0.99	1.51	0.01
Type 2 Fuzzy	0.41	0.98	0.88

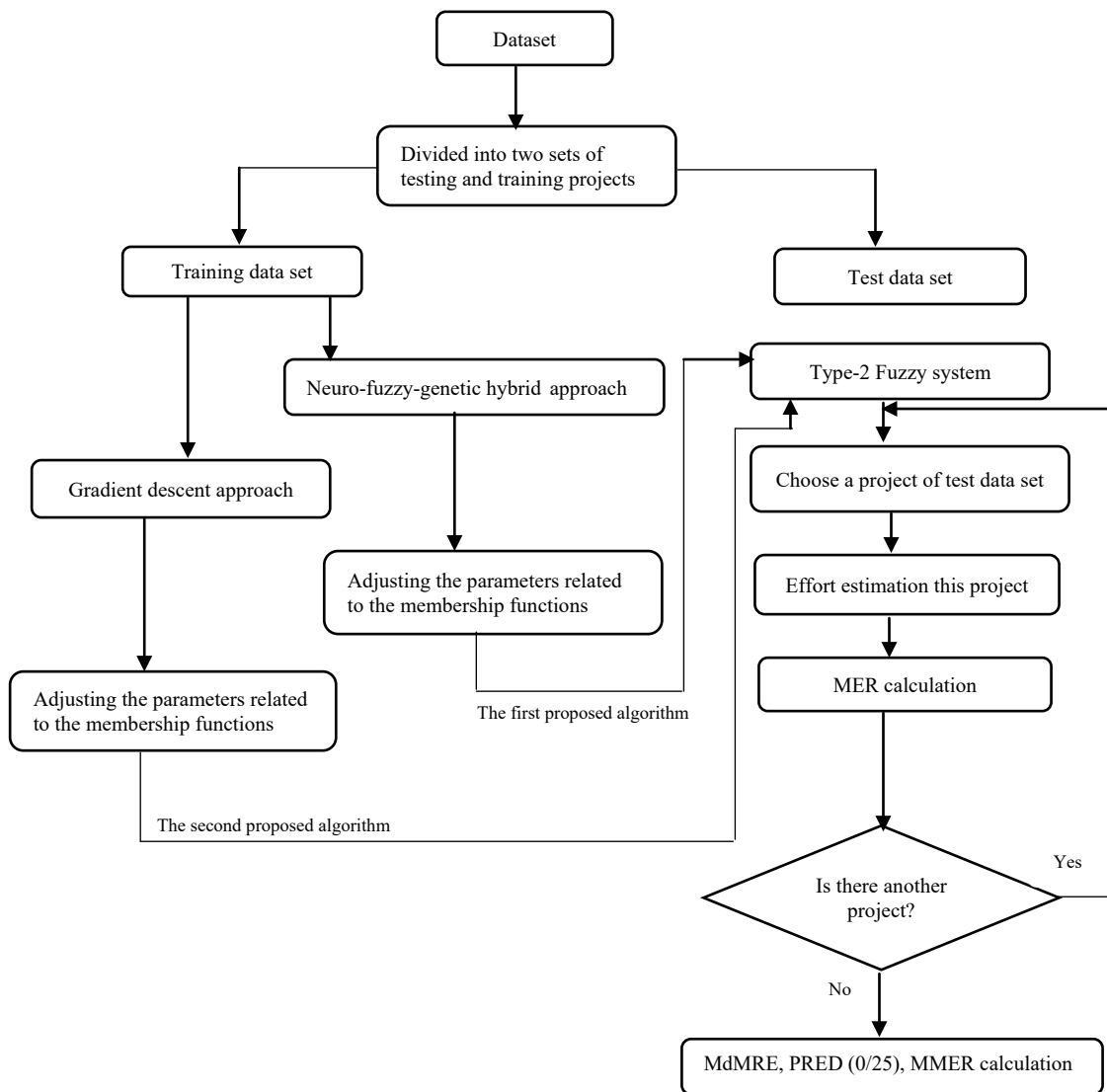
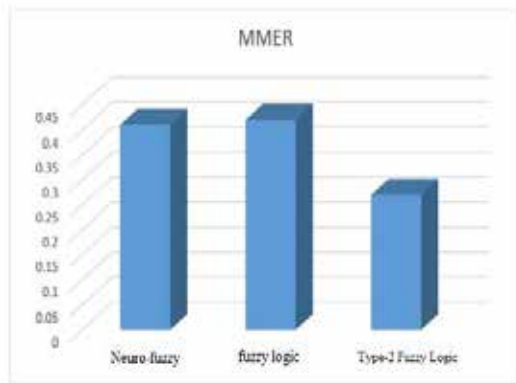
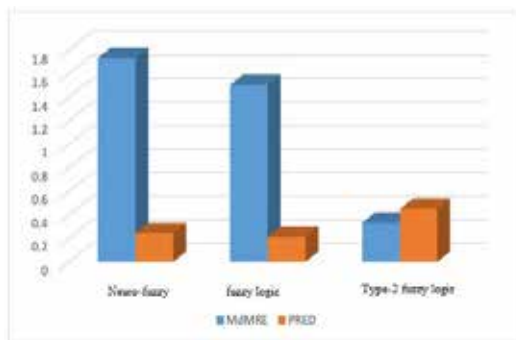


Fig 2: Flowchart of the Proposed Model

As indicated by the results, the proposed model performed well on this database and the obtained accuracy from the proposed model on database is optimal. These results are illustrated in figures 7 and 8.



**Fig 7: Char of different models' MMER results on Desharniz's dataset**



**Fig 8: Char of different models' MdMRE and PRED results on Desharniz's dataset**

percentage have been used. The proposed model has been analyzed through three databases. Considering assessing the results of the accuracy of the proposed model compared to other models based on the three given databases, acceptable results have been obtained from the proposed model. Generally, the proposed model is able to provide optimal accuracy in real projects.

## VI. CONCLUSION

Estimating the effort required for software production and development has been studied for many years and numerous articles have so far been published in this area. Recent statistics suggest the fact that work done in this area does not meet the needs of software developers. In this paper, we have tried to present an efficient model for accuracy improvement and software effort estimation compatibility. The proposed model is based on using type-2 fuzzy logic. In order to teach type-2 fuzzy logic, the gradient descent method and the neuro-fuzzy-genetic hybrid approach have been used. In addition, for assessing the accuracy of the proposed model, three of the most widely used assessment parameters, relative error median, relative error mean, and prediction

## REFERENCE

- [1] Mohanty, S.K., Bisoi, A.K., “Software effort estimation approaches-a review”, *International Journal of Internet Computing*, 2012, 1(3): 82-88.
- [2] Gharehchopogh, F. S. and Z. A. Dizaji (2014). A New Approach in Software Cost Estimation with Hybrid of Bee Colony and Chaos Optimizations Algorithms, MAGNT RESEARCH REPORT.
- [3] Bardsiri, A.K. and S.M. Hashemi, “Software Effort Estimation: A Survey of Well-known Approaches”. *International Journal of Computer Science Engineering (IJCSSE)*, 2014. 3(1): p. 46-50.
- [4] Benala, T. R., et al. (2014). *Software Effort Estimation Using Data Mining Techniques. ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India-Vol I*, Springer
- [5] Heidrich, J., M. Oivo, and A. Jedlitschka, “Software productivity and effort estimation”. *Journal of Software: Evolution and Process*, 2015. 27(7): p. 465-466.
- [6] Elish, M. O., et al. (2013). “Empirical study of homogeneous and heterogeneous ensemble models for software development effort estimation.” *Mathematical Problems in Engineering* 2013.
- [7] Abbas, S.A., et al., “Cost estimation: A survey of well-known historic cost estimation techniques”. *Journal of Emerging Trends in Computing and Information Sciences*, 2012. 3(4): p. 612-636.
- [8] Kumari, S. and S. Pushkar, “Performance analysis of the software cost estimation methods: a review”. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2013. 3(7): p. 229-238.
- [9] Esplanada, P.S. and E.A. Albacea, “Assessing Accuracy of Formal Estimation Models and Development of an Effort Estimation Model for Industry Use”. 2012.
- [10] Ramesh, K. and P. Karunanidhi, “Literature Survey On Algorithmic And Non-Algorithmic Models For Software Development Effort Estimation”. *International Journal Of Engineering And Computer Science ISSN*, 2013: p. 2319-7242.
- [11] Potdar, S. M., et al. (2014). “Literature Survey on Algorithmic Methods for Software Development Cost Estimation.” *International Journal of Computer Technology and Applications* 5(1): 183.
- [12] Basha, S. and D. Ponnurangam, “Analysis of empirical software effort estimation models”. *arXiv preprint arXiv:1004.1239*, 2010.
- [13] Mendel, J. M. “Type-2 fuzzy sets and systems: an overview” *Computational Intelligence Magazine, IEEE* , 2007, 2(1): 20-29.
- [14] Kashyap, S.K. IR and color image fusion using interval type 2 fuzzy logic system. in *Cognitive Computing and Information Processing (CCIP)*, 2015 International Conference on. 2015. IEE.
- [15] Gupta, N. “Comparative study of type-1 and type-2 fuzzy systems”. *Int. J. Eng. Res. Gen. Sci*, 2, 2014, 195-198
- [16] Hassani, H., & Zarei, J. “Interval Type-2 fuzzy logic controller design for the speed control of DC motors”. *Systems Science & Control Engineering*, 2015, 3(1), 266-273
- [17] Singh, R., Kainthola, A., & Singh, T. N. “Estimation of elastic constant of rocks using an ANFIS approach”. *Applied Soft Computing*, 2012, 12(1), 40-45.
- [18] Malathi, S., & Sridhar, S. “Efficient estimation of effort using machine-learning technique for software cost”. *Indian Journal of Science and Technology*, 2012. 5(8), 3194-3196
- [19] Lopez-Martin, Cuauhtemoc, “A fuzzy logic model for predicting the development effort of short scale programs based upon two independent variables *Applied Soft Computing*”, 2011, 724–732.