Cost Estimation Model Using Fifth Generation Language Technique for Software Maintenance Project

Mohammad Islam, Nafees Akhter Farooqui, Syed Ali Mehdi Zaidi, and Mohammad Shafeeq

Abstract Software cost estimation is a standout among the huge demanding tasks in project management for new software. In any case, the procedure of estimation is unsure as it largely relies on certain qualities that are very hazy amid the beginning periods of improvement. This exploration is to give a method to software cost estimation that performs superior to different procedures on the precision of effort estimation. A soft computing procedure has been investigated to beat the vulnerability and error in estimation. This investigation is to expand the constructive cost model by intertwining the possibility of fuzziness into the estimation of size, method of improvement projects, and the cost drivers adding to the general advancement effort. The primary goal of the explorations is to examine the job of the fuzzy inference system method in enhancing the cost estimation precision utilizing COCOMO II by describing inputs variables utilizing fifth GL systems and contrasting their outcomes. The PROMISE dataset is utilized for the assessment of the fuzzy inference system (FIS) procedures. The examinations have been completed utilizing MATLAB simulation conditions.

Keywords Fuzzy logic · Fuzzy inference system · Maintenance cost estimation model · Software cost estimation

M. Islam · N. A. Farooqui (✉)
Department of Computer Science, Era University, Lucknow, Uttar Pradesh, India
e-mail: nafeesf@gmail.com

M. Islam
e-mail: mohdislam3@gmail.com

S. A. M. Zaidi
Department of Computer Science, Shia PG College, Uttar Pradesh, Lucknow, India
e-mail: samzaididr@gmail.com

M. Shafeeq
Department of Computer Science and Engineering, BBD University, Lucknow, Uttar Pradesh, India
e-mail: shafeeq.bbd@bbdu.ac.in

Y. Singh et al. (eds.), *Proceedings of International Conference on Recent Innovations in Computing*, Lecture Notes in Electrical Engineering 1011,
https://doi.org/10.1007/978-981-99-0601-7_40

1 Introduction

Fifth generation language or fifth GL even tends to be programming languages, which contains visual tools to develop a program. It utilizes visual and graphical advancement interface device to make the source language that is assembled with a third GL or fourth GL compiler. Visual programming enables you to see object-oriented structures and drag symbols to gather program squares. There are some important points as follows:

- A fifth-generation programming language is a high level and logic language. User knowledge bases, expert systems, and less programmer control.
- Fifth GL is a programming language dependent on tackling issues utilizing limitations given to the program, as opposed to utilizing a calculation composed by a software engineer.
- Most imperatives-based, logic programming languages, and some decisive languages are fifth GL.
- Fifth GL is utilized mostly in artificial intelligence inquire about.

Artificial intelligence (AI) is growing such a mind-blowing speed; occasionally, it appears to be supernatural. There is a supposition among analysts and designers that AI could develop so monstrously solid that it would be hard for people to control. People created AI frameworks by bringing into them each conceivable they could, for which the people themselves presently appear to be compromised. AI has many other areas as soft computing technologies, neural network, genetic algorithms, fuzzy logic modeling system, etc., for finding the precise expectation of software development cost estimation. The FIS technique has been adopted to predict maintenance cost.

2 Related Work

There is basic two category of models, such as algorithmic and non-algorithmic [1]. Everyone require inputs, a precise estimate of explicit traits, for example, lines of code and other cost drivers like range of abilities which are difficult to procure amid the beginning time of software development. In 1990s, non-algorithmic was conceived to extend estimating cost. Analysts have focus toward novel methodologies that delicate registering, for example, ANN, GA, and fuzzy logic [2, 3]. A portion of early works demonstrates that fuzzy logic offers an amazing etymological interpretation that ready to speak to imprecision in sources of inputs and outputs while giving huge learning ways to deal with model's structure. It is a procedure to take care of issues, which are too complex to be in any way seen quantitatively. It depends on fuzzy-set-theory. It gives a system to speaking to semantic builds, for example, many, low, medium, and high. It gives a deduction structure that empowers fitting man reasoning limits. Unexpectedly, the binary set hypothesis depicts crisp events that do or do not happen. This experiments perspective hypothesis that clarify if the events will happen evaluating the opportunity for which the given events are required happening [4, 5].

It is the rule-based system which is the core learning containing the implied Fuzzy IF THEN guidelines in which a couple of words are portrayed by consistent part works. It can be classified into three kinds: pure, Takagi and Sugeno's fuzzy logic with fuzzifiers and defuzzifiers. The enormous bit for planning produces crisp-data as for input and envisions crisp-data as for output. This was right off the bat created by Mamdani. It has been viably associated with a collection of mechanical techniques and customer things [6, 7].

It is the initial phase in the fuzzy inference process. This includes an area change where crisp inputs sources are changed into fuzzy inputs. Crisp inputs are careful information sources estimated by sensors and go into the control system for handling, for example, temperature, weight, etc. [8–10]. Each crisp input that will be prepared by the FIS has its gathering of membership functions or sets to which they are changed. This gathering of membership functions exists inside a vast expanse of talk that holds every single important value that the crisp input can have. The accompanying demonstrates the structure of membership functions inside a vast expanse of talk for a crisp input [11, 12].

The principal of FIS is concentrated at the capacity of fuzzy logic that show characteristic. This system contains fuzzy-rules worked for expert-knowledge and called fuzzy expert systems, contingent upon their last use. Before FIS, it was at that point applied to construct expert systems for recreation objectives [13, 14]. The master frameworks depended upon the classical-boolean-logic that was not appropriate for dealing with the sequentially to the fundamental procedure wonders. Fuzzy-logic enables continuous standards that be brought into expert-knowledge-based test systems [7, 15].

The Sugeno's initial tasks, a great deal of scientists, have been engaged with structuring fuzzy systems from databases. The means of fuzzy reasoning performed by FISs are as follows:

1. Comparison of input factors with the MF on the precursor portion to get the membership estimations of each phonetic mark. (Progression is frequently said fuzzification).
2. Connection of the membership values on the reason portion to get terminating quality (level of satisfaction) of each standard.
3. Production of certified ultimately (either fuzzy or crisp) or each standard relying upon terminating quality.
4. Composite the certified consequents to deliver a crisp output. (Progression is said defuzzification).

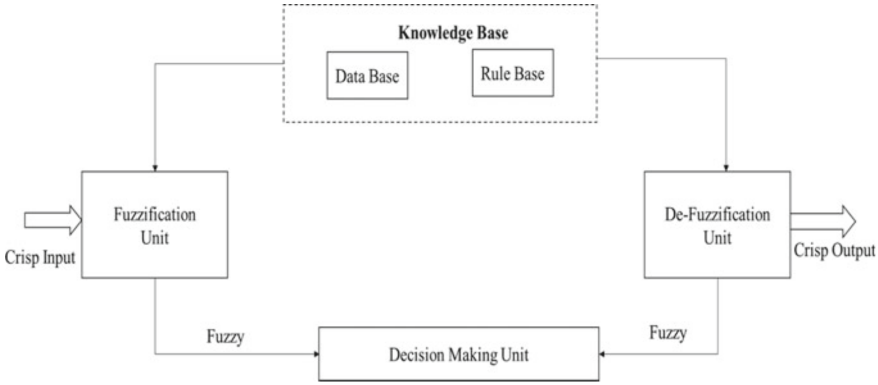

Fig. 1 Fuzzy inference system (FIS)

3 Research Methodology Used

The FIS is used to execute the differing preparing propels. Decisions were obliged making and adjusting FIS with fuzzy logic toolbox software using graphical instruments or command line capacities. The research will implement on third GL, fourth GL, and fifth GL using Mamdani FIS. Figure 1 is used as fuzzification and defuzzification.

The performance analysis and their corresponding results are compared. The results are analyzed using the criterion RMSE. Less value means that the result is more accurate.

- Select a specific kind of FIS (Mamdani).
- Define the variables for the input and output.
- Set input and output member functions.
- The data is now in rule editorial manager if-then rules.
- An explicit model structure is made, and parameters of input and output factors tuned to get the ideal output.

4 Proposed Model for Software Maintenance Cost Estimation

COCOMO II is utilized as the model-based to assess the software project cost. The model was developed by Mr. Boehm and Scattered in 1981 that utilizing aggregated data from 63 projects. It is a good manual that estimating maintenance cost for software. This proposed model implies with new factual methodologies and strategies that estimate the maintenance cost of software using fifth GL (fuzzy inference system) procedure.

The issue of software cost estimation is that everything considered and relies on single estimations of size, cost drivers, and scale factors. It is assessed dependent on recently finished projects that are fairly like the present projects [16–18]. Similarly, cost drivers and scale elements need through evaluation instead of doling out a fixed numbers value. To beat this condition, it is more brilliant to address these responsibilities to the kinds of fuzzy-sets, where the qualities of interval are utilized which is expressed through collection of membership functions like triangular MF, trapezoidal MF, and Gaussian MF [19].

The proposed fuzzy software cost estimation model is represented in the Fig. 2. Its principles contain phonetic factors identified with the undertaking. The FIS utilizes connecters "and/or" for COCOMO input factors that shape principles. The FIS incorporates many input software characteristics: seventeen cost drivers, five scale factors, one size (KDLOC), and one output as cost estimation (CE).

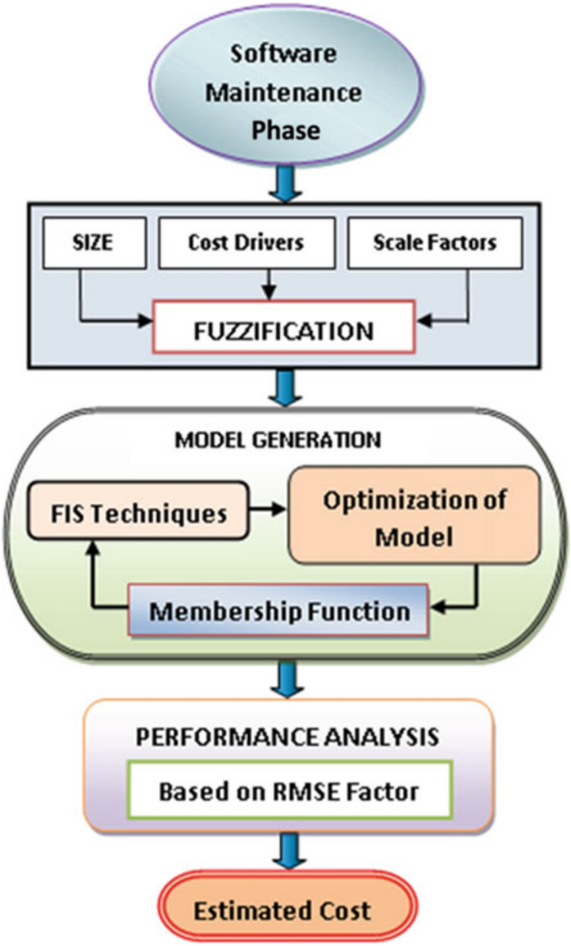

Fig. 2 Proposed maintenance cost estimation model using FIS techniques

Fuzzy set takes all the input and convert into the phonetic values. For each cost driver, a different FIS is planned. Principles are created as cost drivers for forerunner parts that comparing effort-multiplier in the resulting portion. The defuzzified value for the effort-multiplier has kept for separate FIS. The scale factors are additionally fuzzified. The pcap means that programmer-capability is examined for an example. Programmer-capability for fuzzification depends on COCOMO II, calibrated model of post-architecture qualities.

Next, the model so obtained will be later subjected to optimization of its model parameters using fuzzy inference system optimization technique to arrive at better software cost estimation prediction accuracy. The fuzzy operators such union, intersection, and complement shall be used. FIS races to create answers for progressive generations. Henceforth, the quality of the solutions in progressive generations improves. The procedure is ended when an ideal solution is found. The result has analyzed the criterion of root mean square error (RMSE) factor, which predicts the better software cost estimation with accuracy.

4.1 Data Used for Validation

The data used as input and output variables for ideal COCOMO II model advancement is given in Table 1. The dataset Table 2 is assembled from the examination of 40 software projects, which is adopted from Software Engineering Repository of PROMISE dataset which open access for researching reason. It comprises 26 attributes like seventeen standards COCOMO II characteristics cost drivers and five scale factors in the range that measure in thousand delivered source lines of code (KDLOC) directions. The output of the model is the cost estimation (CE), measured in man-months. The estimated efforts using third GL, fourth GL, and fifth GL approaches obtained are tabulated and compared. The model equation is given as follows:

$$\mathrm{PM} = A \times [\mathrm{Size}]^{1.01} + \sum_{i=1}^{5} \mathrm{SF}i \times \prod_{i=1}^{17} \mathrm{EM}i \quad (1)$$

Here, effort is indicated in terms of person-months (PM), A is a constant that is multiplicative, size is the projected-size of the software that expressed in KDLOC, EMi ($i = 1, 2, 3, 0.17$) are effort-multipliers, and SFi ($i = 1, 2, 3, 0.5$) are scale factors as exponent. It is a specific normal for product improvement that has impact that increments or decrements the measure for advancement effort [20]. There are team cohesion, process maturity, architecture/risk resolution, development, flexibility, and precedent Ness. All the effort-multipliers are gathered into 4 parts, which are project-factors, personnel, platform, and product. The items are utilized to modify the nominal effort.

Table 1 COCOMO II effort multipliers [5]

Input	Variables	Cost drivers and scale factors
1	"ACAP"	Analyst capability
2	"APEX"	Applications-experience
3	"CPLX"	Product-complexity
4	"DATA"	Size of database
5	"DOCU"	Documentations (life cycle need)
6	"FLEX"	Languages and tool-experiences
7	"PCAP"	Capability of programmer
8	"PCON"	Personnel continuity
9	"PERS"	Personnel capability
10	"PLEX"	Platform-experience
11	"PMAT"	Process-maturity-level (equivalent)
12	"PREC"	Precedentness for applications
13	"PREX"	Personal-experiences
14	"PVOL"	Volatility platform
15	"RELY"	Software-reliability (required)
16	"RESL"	Risk-resolution
17	"SITE"	Multisite-development
18	"STOR"	Main-storage (constraint)
19	"TEAM"	Team-cohesion
20	"TIME"	Time-execution (constraint)
21	"TOOL"	Software tools used
22	"SIZE"	SS
Output	Variables	Cost estimation (CE)

4.2 *Fuzzy Inference System Rules Applied*

Fuzzy principles for the fuzzy inference system dependent on COCOMO II are characterized with semantic factors in the fuzzification procedure. These principles depend upon connective "and" as between the input factors.

The Rules are defined as follows:
  if (rely is vl) then (effort is vl)
  if (rely is l) then (effort is l)
  if (prec is vl) then (effort is xh)
  if (pmat is vh) then (effort is l)

The following rules are used in Figs. 3, 4, and 5:
  if (pcap is very low) then (increased effort)
  if (pcap is low) then (increased effort)
  if (pcap is nominal) then (unchanged)

Table 2 Estimated-effort using that different MFs

RMSE using different generation languages approaches					
P. no	Actual	COCOMO II	Third GL	Fourth GL	Fifth GL
1	2040	2215.24	2089	2075	1945
2	321	213.83	201.6	200.4	187.8
3	79	107.85	102	101.15	94.35
4	6550	7806.45	7372	7329.2	6854.7
5	724	733.16	690.9	687.14	643.9
6	121.6	149.6	166.68	165.25	125.90
7	117.6	144.9	98.87	99.39	114.13
8	33.2	42.9	42.56	42.79	36.27
9	36	40.1	48.10	48.36	36.64
10	35.2	41.5	39.79	40.01	37.38
11	8.4	18.4	6.12	5.65	5.24
12	10.8	19.3	3.63	3.65	13.81
13	352.8	369.9	444.90	447.09	340.60
14	70	81.9	64.98	65.83	63.44
15	72	79.8	45.32	45.71	61.45

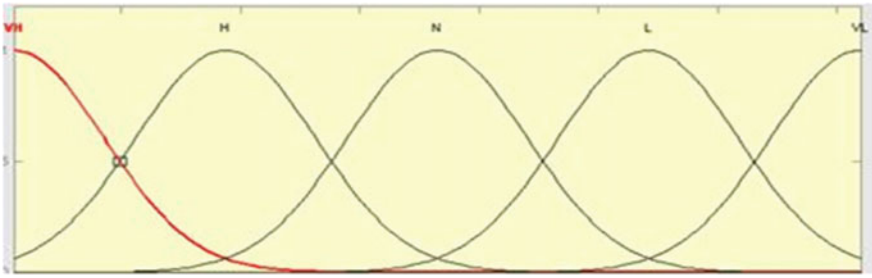

Fig. 3 PCAP fuzzification cost drivers using Gaussian MF

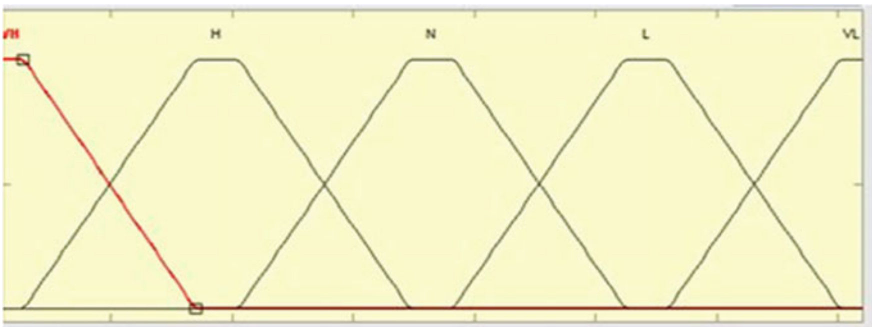

Fig. 4 PCAP fuzzification cost drivers using trapezoidal MF

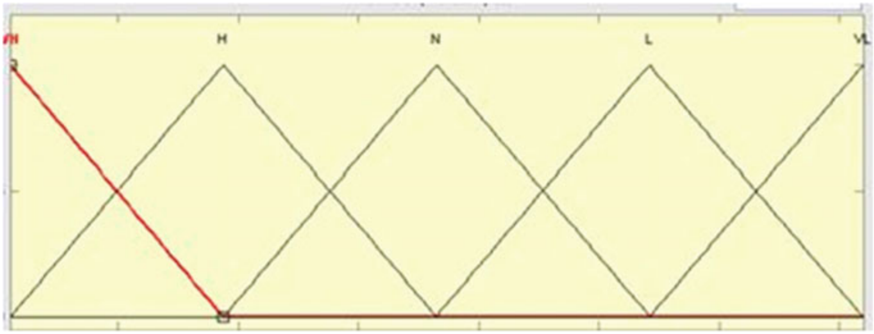

Fig. 5 PCAP fuzzification cost drivers using triangular MF

if (pcap is high) then (decreased effort)
if (pcap is very high) then (decreased effort)

Figure 6 expresses the graphical user interface that developed our model FIS. We can legitimately enter the qualities and get the relating effort. The studies have been carried out using MATLAB simulation environment.

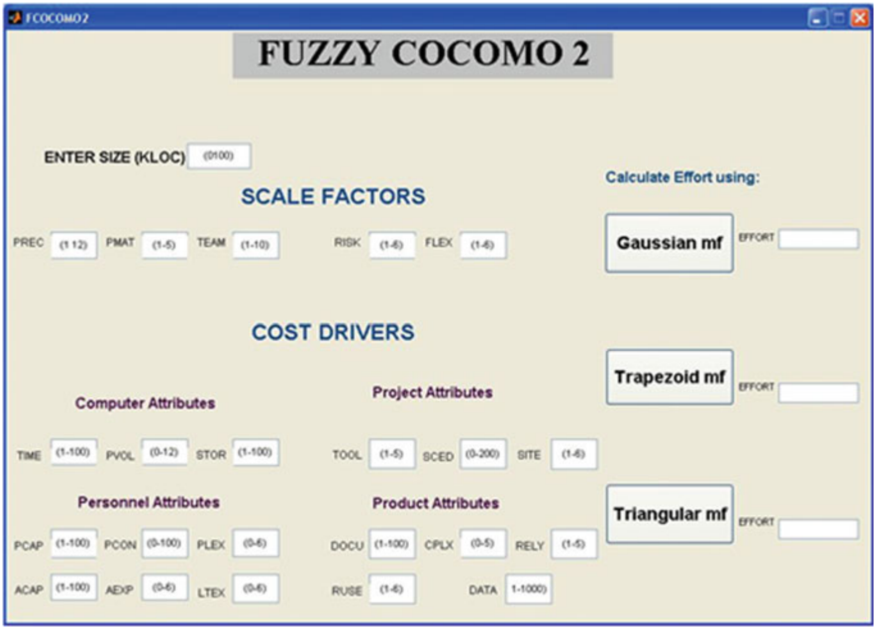

Fig. 6 Interface used for cost evaluation

5 Experimental Results

5.1 Root Mean Squared Error (RMSE)

The assessment comprises in contrasting the exactness of the calculated cost with genuine cost. There are numerous assessment scales for estimating software cost. We connected the regular one is RMSE and defined as follows:

$$\mathrm{RMSE} = \left(\frac{1}{N}\sum_{i=1}^{N}\left(y_i - \overset{\wedge}{\underset{i}{y}}\right)^2\right) \tag{2}$$

5.2 Mean Absolute Error (MAE)

It is a proportion of expectation precision of a forecasting method in insights, for instance, in pattern estimation. The most part communicates precision as a rate and is characterized by the equation:

$$M = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \tag{3}$$

The software effort got when utilizing regular COCOMO II and fuzzy MF were looked at. In the wake of breaking down, the outcomes acquired utilizing applying third, fourth, and fifth GL. It is shown that the cost evaluated by fuzzifying all effort-multipliers utilizing fifth GL (FIS) procedure is predicting better estimate.

6 Comparison

The parameter of cost estimation models for the assessment is the MAE that is represented in the Eq. 3. The effort has been calculated for every observation (Table 3).

Table 4 demonstrates with the chart representing to the similar examination of the real cost with the estimated cost using COCOMO II, third, fourth, and fifth GL. The RMSE and MAE values are calculated using Eqs. 2 and 3, and the RMSE values for all project for COCOMO II, third GL, fourth GL, and fifth GL are 1.2403, 1.0638, 1.075, and 0.9398, respectively. The MAE values are 0.1650, 0.1251, 0.1236, and 0.1183, respectively. This plainly demonstrates here is a reduction in the absolute errors; therefore, the proposed model is progressively reasonable for estimating cost.

Table 3 Comparison of MAE values

MAE using different generation languages approaches				
P. no	COCOMO II	Third GL	Fourth GL	Fifth GL
1	0.0859	0.024	0.0171	0.0465
2	0.3339	0.3719	0.3757	0.4149
3	0.3651	0.2911	0.2803	0.1943
4	0.1828	0.1171	0.1105	0.0385
5	0.0126	0.0457	0.0509	0.1106
6	0.2242	0.1531	0.1462	0.0746
7	0.0825	0.072	0.017	0.0451
8	0.1533	0.0866	0.0813	0.0113
9	0.4104	0.0630	0.064	0.0392
10	0.3651	0.2911	0.2803	0.1943
11	0.1828	0.1171	0.1105	0.0385
12	0.0126	0.0457	0.0509	0.1106
13	0.2242	0.1531	0.1462	0.0746
14	0.0815	0.020	0.017	0.0491
15	0.152	0.0846	0.0817	0.0121

Table 4 Comparison of RMSE and MAE factors

Comparison of cost estimation techniques	RMSE	MAE
COCOMO II model	1.2403	0.1650
Third GL model	1.0638	0.1251
Fourth GL model based on CBSD approaches	1.075	0.1236
Fifth GL model based on FIS technique	0.9398	0.1183

7 Conclusion

We conclude that the use of fuzzy logic in SCE yields more exact outcomes than the past experimental model methodology. The RMSE values of cost estimation using fifth GL based on FIS techniques give better outcomes for most extreme rules if other high-level language techniques will be used. It found that the FIS is achieving better as it shows a likely change in its intervals, and accomplished outcomes were nearer to the actual cost. FIS has the lowest MAE and highest accuracy of the three generation language software methodologies that studied.

Future work incorporates more current procedures, i.e., type-2 fuzzy-system can likewise be connected for increasingly precise forecasts of software. The above research work can be easily employed in the software industries.

References

1. Lagerström R, von Würtemberg LM, Holm H, Luczak O (2016) Identifying factors affecting software development cost and productivity. Softw Qual J 20:395–417
2. El Bajta M, Idri A, Fernandez-Alem JL, Ros JN (2015) Software cost estimation for global software development. In: 10th International Conference on Evaluation of Novel Software Approaches to Software Engineering, ENASE
3. Veeranjaneyulu N, Suresh S, Salamuddin S, Kim H-Y (2014) Software cost estimation on e-learning technique using a classical fuzzy approach. Int J Soft Eng Appl 8(11):217–222
4. Patil LV, Waghmode RM, Joshi SD, Khanna V (2017) Generic model of software cost estimation: A hybrid approach. IEEE Int Adv Comput Conf, IEEE, pp 1379–1384
5. Vu N (2010) Improved size and effort estimation models for software maintenance. University of Southern California
6. Marounek P (2012) Simplified approach to effort estimation in software maintenance. University of Economic, Prague, Faculty of Information and Statistics. J Syst Integration, 51–63
7. Wijayasiriwardhane T, Lai R, Kang KC (2011) Effort estimation of component based software development, a survey. IET Soft 5:216–228
8. Borade JG (2013) Software project effort and cost estimation techniques. Int J Adv Res Comp Sci Soft Eng 3(8):730–739
9. Chiu S (1994) Fuzzy model identification based on cluster estimation. J Intell Fuzzy Syst, 267–278
10. Farooqui NA, Ritika (2020) A machine learning approach to simulating farmers' crop choices for drought prone areas. In: Singh P, Panigrahi B, Suryadevara N, Sharma S, Singh A (eds) Proceedings of ICETIT 2019. Lecture Notes in Electrical Engineering, vol 605. Springer, Cham. https://doi.org/10.1007/978-3-030-30577-2_41
11. Aggarwal K, Singh Y, Chandra P, Puri M (2005) Measurement of software maintainability using a fuzzy model. J Comp Sci. ISSN 1549-3636
12. Sneed H (2004) A cost model for software maintenance & evolution. IEEE, 264–273
13. Patil LV, et al (2014) Develop efficient technique of cost estimation model for software applications. Int J Comp Appl 87(16):0975–8887, February
14. Maleki I, Ebrahimi L, Jodati S, Ramesh I (2014) Analysis of software cost estimation using fuzzy logic. Int J Foundations of Comp Sci Tech 4(3):27–41
15. Mukherjee S, Bhattacharya B, Mandal S (2013) A survey on metrics, models & tools of software cost estimation. Int J Adv Res Comp Eng Tech (IJARCET) 2(9):2620–2625
16. Jing QF, Zhu X-Y, Xiaoyuan XX, Baowen X, Shi Y (2017) Software effort estimation based on open-source projects: case study of Github 92:145–157, December
17. Patil PK (2015) A review on calibration factors in empirical software cost estimation (SCE) models. Int J Soft Eng Res Pract 3:1–7
18. Mohammad I, Vinodani K (2014) Development of a software maintenance cost estimation model: 4TH GL perspective. Int J Tech Res Appl, 6
19. Sehraab SK, Brara YS, Sukhjit NK, Sehra S (2017) Research patterns and trends in software effort estimation 91:1–21, November
20. Srivastava SK, Prasad P, Varma SP (2016) Evolving predictor variable estimation model for Web engineering projects. Comput Sci Eng, 68–89