A Fuzzy Logic Based Software Development Cost Estimation Model with improved Accuracy

Shrabani Mallick[1] and Dharmender Singh Kushwaha[2]

[1] Dr. B. R. Ambedkar Institute of Technology, Port Blair, A & N Islands, India, [2]Motilal Nehru National Institute of Technology, Allahabad, Uttar Pradesh, India

Abstract
Software cost and schedule estimation is usually based on the estimated size of the software. Advanced estimation techniques also make use of the diverse factors viz, nature of the project, staff skills available, time constraints, performance constraints, technology required and so on. Usually, estimation is based on an estimation model prepared with the help of experienced project managers. Estimation of software cost is predominantly a crucial activity as it incurs huge economic and strategic investment. However accurate estimation still remains a challenge as the algorithmic models used for Software Project planning and Estimation doesn't address the true dynamic nature of Software Development. This paper presents an efficient approach using the contemporary Constructive Cost Model (COCOMO) augmented with the desirable feature of fuzzy logic to address the uncertainty and flexibility associated with the cost drivers (Effort Multiplier Factor). The approach has been validated and interpreted by project experts and shows convincing results as compared to simple algorithmic models.
Keywords:
COCOMO, fuzzy logic, software development, cost estimation.

1 Introduction

The earliest estimation models, for software cost estimation *the Basic COCOMO,* used single variable (i.e, software size) static estimation based on the type of the software for estimating the development effort. Advanced variants of Basic COCOMO models - Intermediate and Detailed COCOMO, referred hence forth as COCOMO-Iprovides subjective estimations based on the size of software and a set of other parameters called the cost directives categorised into 4 categories:

1. Product attributes
2. Hardware attributes
3. Personnel attributes
4. Project attributes

These attributes tries to take into account the dynamics of the software development that can affect the cost and effort required for developing the software as enlisted in Table 1.

Manuscript received June 5, 2024
Manuscript revised June 20, 2024
https://doi.org/10.22937/IJCSNS.2024.24.6.3

Table 1: Cost Directives of intermediate COCOMO

Product	-Required software reliability -Size of application database -Complexity of the product
Hardware attributes	-Run-time performance constraints -Memory constraints -Volatility of the virtual machine - environment -Required turnabout time.
Personnel attributes	-Analyst capability -Software engineering capability -Application experiences -Virtual machine experience -Programming language experience
Project attributes	-Use of software tools -Application of software engineering methods Required development schedule.

The scale of severity of these cost directivesvaries on a scale of 1 to 6 ranging from Very Low to Prominent as shown in Fig 1. Based on the severity of each of the cost directives, an effort multiplier factor (EMF) has been assigned to it in a range of 0.9 to 1.9 which are hypothetical derived from historical analysis of various projects.

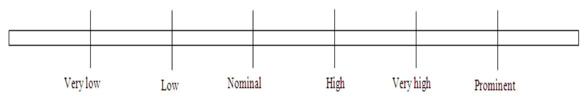


Fig 1: Scales of severity of cost directives

Most of the work in the cost estimation field has focused on algorithmic cost modelling. In this process costs are analysed using mathematical formulas linking costs or inputs with metrics to produce an estimated output. The formulae used in a formal model arise from the analysis of historical data. All real life situations particularly a complex engineering task like software development have some degree of fuzziness. Thus the accuracy of the cost model as compared to actual cost can be improved by converting the crisp values of EMF to fuzzy sets superimposed by fuzzy logic rules of the form IF-THEN that convert input fuzzy set to output fuzzy set and then defuzzified to give a near correct estimate.This paper presents a fuzzy logic based flexible and efficient Constructive Cost Model (COCOMO) to increase the accuracy of software development cost prediction.

2 Related Works

Putnam's [Putnam 78][8], SLIM is one of the first algorithmic cost model. It is based on the Norden/Rayleigh function and generaly known as a macro estimation model. It is primarily used for large projects.Boehm's [Boehm 81][9],COCOMO model is one of the mostly used model commercially. The first version of the model delivered in 1981 followed by COCOMO II . In [1], authors propose a modelthat carries some of the desirable features of neural networks approach, such as learning ability and good interpretability, while maintaining the merits of the COCOMO model, In [2, 4] the authors utilise an adaptive fuzzy logic model for better accuracy of software time and cost estimation using a Two-Dimension Gaussian Membership Function (2-D GMF). [3], presents an analytical structure of a Takagi-Sugeno fuzzy logic controller with three inputs and one output for software development effort estimation. In [5], the authors present a neuro-fuzzy inference system to handle the dependencies among contributing factors and decouple the effects of the contributing factors into individuals to predict software cost. In [6], the authors provide an objective, reproducible and quantitative measure for software project management and applying Fuzzy Logic approach to the software metrics modeling. In [7] the authors present a software cost estimation model based on fuzzy logic. The fuzzy logic model fuzzifies the two parts of the COCOMO model i.e. nominal effort prediction and the effort adjustment factor. The analysis shows that the performance of the FIS enhanced by increasing the number of membership functions. [10] The main goal of this research was to design and compare three different fuzzy logic models for predicting software estimation effort: Mamdani, Sugeno with constant output, and Sugeno with linear output. Fuzzy logic models were found to be very sensitive to outliers. Authors in [12], present a novel neural network Constructive Cost Model (COCOMO) is proposed for software cost estimation. This model carries some of the desirable features of neural networks approach, such as learning ability and good interpretability, while maintaining the merits of the COCOMO model. The authors [11] propose a combined model of cost estimation using COCOMO and Gaussian Membership Function to minimize the relative error. A fuzzy-based analogy is obtained in the present study to select the nearest path from the history available to meet the project cost and time.

The work proposed in this paper analyses the effect of change of cost drivers elicited from Intermediate COCOMO model and estimate the impact based on Detailed COCOMO in a phase wise distributed manner. This is then mapped to the Fuzzy Inference System. The novelty of the work lies on the fact that it has attempted to calculate the change factor in cost estimation given a set of changed requirement after the initial estimate has been done.

3 Approach

Project managers can use algorithmic cost model to analyse and compare different ways of investing money to reduce project costs. As discussed in table 1, the key cost drivers fall into four classes, namely product attributes (PRODA), Hardware attributes(HWDA), personnel attributes(PERSA) and project attributes(PROJA). These factors decide the cost of a software on and above a fixed basic cost multiplied by the average person-month of effort decided by the organization. As per the COCOMO, the software development cost SC is calculated as follows:

$$SC = Basic\ cost\ x\ E\ x\ D\ x\ P \quad (1)$$

$$Where\ Effort = E = a_i(KLOC^{b_i})x\ EAF \quad (2)$$

$$Duration = D = c_b(E^{d_b}) \quad (3)$$

$$Person\ Deployed = P = E/D \quad (4)$$

a_i, b_i, c_b and d_b are constant factors of COCOMO. EAF is an effort adjustment multiplier that is dependent on the cost drivers PRODA, HWDA, PERSA and PROJA called the Effort Multiplier Factors (EMFs). The EMFs vary from 0.9 to 1.4.

The algorithmic approach works fairly well when the requirements, feasibilities and constraints are clearly fixed before-hand based on which a fair estimate is prepared. The problem arises when the software requirements are not very clear and is liable to change or very dynamic in nature. Changing requirement or constraints means change in the effort multipliers (EMFs) which certainly impacts the projected cost of the software. This entails frequent analysis of the cost change over the pre-estimated cost to see the change impact so as to keep track of hardware/software/development cost trade-offs. The issues that arise are elicited as follows:

1. The change in EMFs may affect various phases of the life cycle.
2. The effect of change is distributed across various phases of the life cycle which is fuzzy in nature.
3. What is the factor by which the cost is getting affected given a set of changed attributes required for the project?

The paper addresses these issues as:

1. Establishing a correlation between Attribute and Life Cycle Phase as per the detailed COCOMO model, so as to understand the maximum impact of a particular attribute.
2. Changing the crisp set of EMFs that have values ranging from 0.9 to 1.4 according to the COCOMO model, to fuzzy sets to capture the fuzziness and support flexible sense of memberships defined for a value.
3. Creating a rule-based Fuzzy Inference Systemfor estimating the change factor (CF), by which projected cost may change given a set of changed attributes. This factor is used as a multiplier to calculate the change impact on a pre-estimated cost. The Rule-base can be interpreted and validated by project experts and can fine-tuned based on their past experience. Thus the new Software development cost (NSC) is given as:

$$NSC=SC*CF \tag{5}$$

The subsequent section gives the implementation aspects that address the above issues.

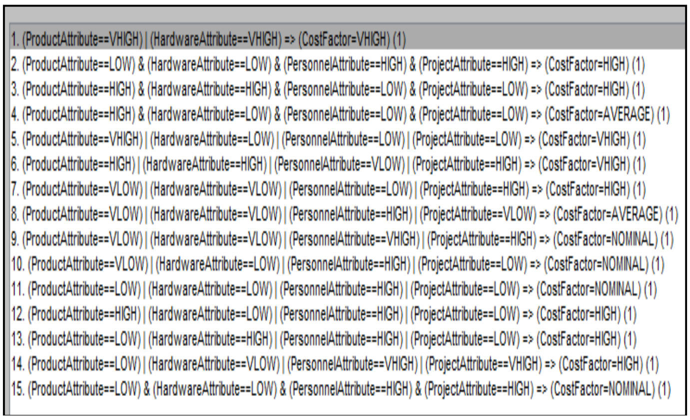
4 Phase wise Change Effect Distribution

To provide a close estimation for finding the change impact of a cost driver attribute on the estimated cost, a phase wise impact analysis is done. Software development is executed in phased manner. Although the phases are seamless but each phase addresses a particular scope of work. In tune with the phase-wise effort distribution of the Detailed COCOMO, the change impact across the phases has been assessed as the change is proportional to the amount of effort. Table 2 enumerates the Change Impact Distribution of the cost drivers in medium sized semi-detached projects.

Table 2 Change Impact Distribution of the cost drivers

Phase→ Cost Drivers ↓	Project planning	Requirements Analysis	Designing	Coding & Unit Testing	Integration and System Test
Product Attributes	10%	10%	60%	20%	--
Hardware Attributes	--	20%	30%	50%	--
Personnel Attributes	--	20%	30%	40%	10%
Project Attributes	10%	20%	20%	30%	20%

According to detailed COCOMO the maximum effort consumption phases are the Designing and Code Writing. As can be seen from Table 2, the change in requirement of Product, Hardware and Personnel attributes impacts the cost the most which forms the basis of the Rule-base. An illustrative Rule-base is shown in Fig 2.



```
1. (ProductAttribute==VHIGH) | (HardwareAttribute==VHIGH) => (CostFactor=VHIGH) (1)
2. (ProductAttribute==LOW) & (HardwareAttribute==LOW) & (PersonnelAttribute==HIGH) & (ProjectAttribute==HIGH) => (CostFactor=HIGH) (1)
3. (ProductAttribute==HIGH) & (HardwareAttribute==HIGH) & (PersonnelAttribute==LOW) & (ProjectAttribute==LOW) => (CostFactor=HIGH) (1)
4. (ProductAttribute==HIGH) & (HardwareAttribute==LOW) & (PersonnelAttribute==LOW) & (ProjectAttribute==LOW) => (CostFactor=AVERAGE) (1)
5. (ProductAttribute==VHIGH) | (HardwareAttribute==LOW) | (PersonnelAttribute==LOW) | (ProjectAttribute==LOW) => (CostFactor=VHIGH) (1)
6. (ProductAttribute==HIGH) | (HardwareAttribute==HIGH) | (PersonnelAttribute==VLOW) | (ProjectAttribute==HIGH) => (CostFactor=VHIGH) (1)
7. (ProductAttribute==VLOW) | (HardwareAttribute==VLOW) | (PersonnelAttribute==LOW) | (ProjectAttribute==HIGH) => (CostFactor=HIGH) (1)
8. (ProductAttribute==VLOW) | (HardwareAttribute==VLOW) | (PersonnelAttribute==HIGH) | (ProjectAttribute==VLOW) => (CostFactor=AVERAGE) (1)
9. (ProductAttribute==VLOW) | (HardwareAttribute==VLOW) | (PersonnelAttribute==VHIGH) | (ProjectAttribute==HIGH) => (CostFactor=NOMINAL) (1)
10. (ProductAttribute==VLOW) | (HardwareAttribute==LOW) | (PersonnelAttribute==HIGH) | (ProjectAttribute==LOW) => (CostFactor=NOMINAL) (1)
11. (ProductAttribute==LOW) | (HardwareAttribute==LOW) | (PersonnelAttribute==HIGH) | (ProjectAttribute==LOW) => (CostFactor=NOMINAL) (1)
12. (ProductAttribute==HIGH) | (HardwareAttribute==LOW) | (PersonnelAttribute==HIGH) | (ProjectAttribute==LOW) => (CostFactor=HIGH) (1)
13. (ProductAttribute==LOW) | (HardwareAttribute==HIGH) | (PersonnelAttribute==HIGH) | (ProjectAttribute==LOW) => (CostFactor=HIGH) (1)
14. (ProductAttribute==LOW) | (HardwareAttribute==VLOW) | (PersonnelAttribute==VHIGH) | (ProjectAttribute==VHIGH) => (CostFactor=HIGH) (1)
15. (ProductAttribute==LOW) & (HardwareAttribute==LOW) & (PersonnelAttribute==HIGH) & (ProjectAttribute==HIGH) => (CostFactor=NOMINAL) (1)
```

Fig 2: Rule Base for the Fuzzy Inference System

The Rule Quantifiers {VLOW, LOW, HIGH, VHIGH} for the different inputs are to be inferred as {NOMINAL*, AVERAGE*, HIGH*, VHIGH*} on the output parameter as shown in Table 3

Table 3 Input-Output inference map

Cost Drivers	V.LOW	LOW	HIGH	V.HIGH
Product Attributes	NOMINAL*	AVERAGE*	HIGH*	V.HIGH*
Hardware Attributes	NOMINAL*	NOMINAL*	AVERAGE*	V.HIGH*
Personnel Attributes	V.HIGH*	HIGH*	NOMINAL*	NOMINAL*
Project Attributes	NOMINAL*	NOMINAL*	HIGH*	V.HIGH*

The starred entries in Table 2 are quantifiers for the output parameter Cost Change Factor, which is defined over [0 – 1].

The next section discusses the proposed Fuzzy Inference System in greater detail by characterizing the Fuzzy Inference System (FIS) model.

5 Estimating the Change Factor using the proposed Fuzzy Inference System

Logic is the science of reasoning. The Fuzzy Inference System has turned out to be an effective for inferring and deducing information from a given set of facts using fuzzy logic. Just as the crisp logic is built on a 2-state membership (0/1), fuzzy logic is built on a multistate truth value. The proposed model captures the various cost drivers discussed in the introductory section and converts them into fuzzy sets. The inputs are taken as Product Attribute, Hardware Attribute, Personnel Attribute and Project Attribute whereas the output is taken as the cost factor, Fig 3. The fuzzy set which characterizes the inputs is as given in Fig 4.The Fig 5 shows the output fuzzy set.

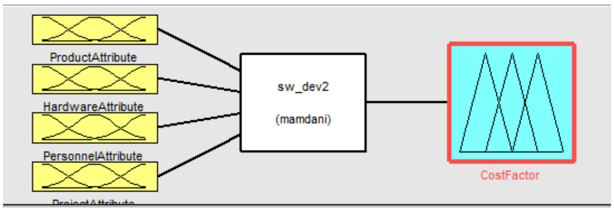


Fig 3 The Proposed Fuzzy Inference System model

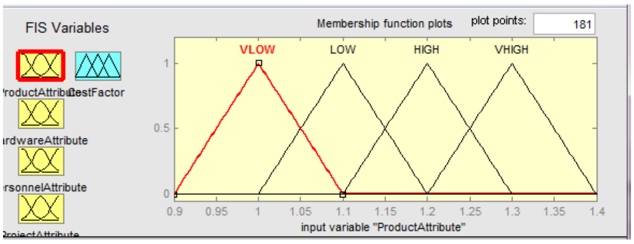


Fig 4 Product Attribute (normalized)

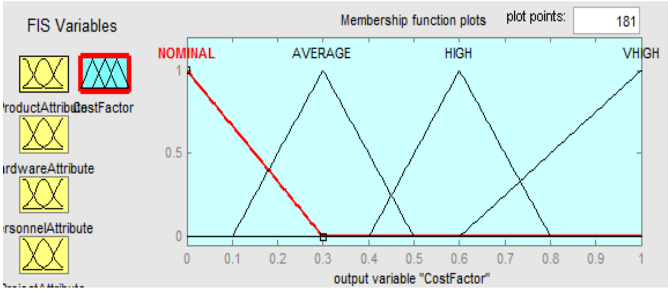


Fig 5 Cost Change Factor (CF) (normalized)

A sample output is shown in Figure 6, which shows that if the change in product attribute is high, Hardware Attribute is High, Personnel Attribute is VLow and Project Attribute is High then the cost increases almost by a factor of 0.543.

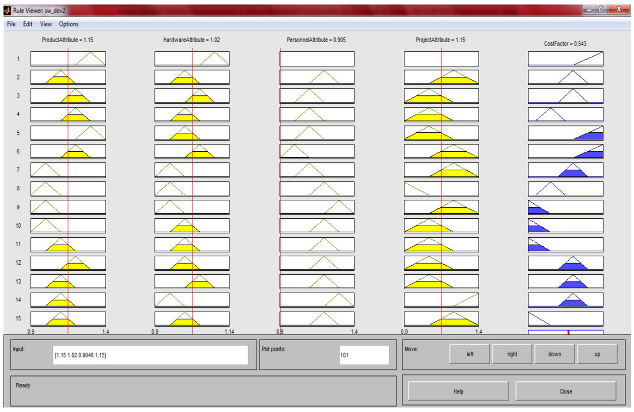


Fig 6. A sample Case – Correlation between input and output parameters

The results were validated for 10 sample projects of roughly 32,000 LOC(medium sized projects) with various cost driver scenario and the cost estimated though COCOMO were compared with the proposed model and the statistics show convincing results, as shown in Table 4.

Table 4 Data from various projects for validation of Proposed model

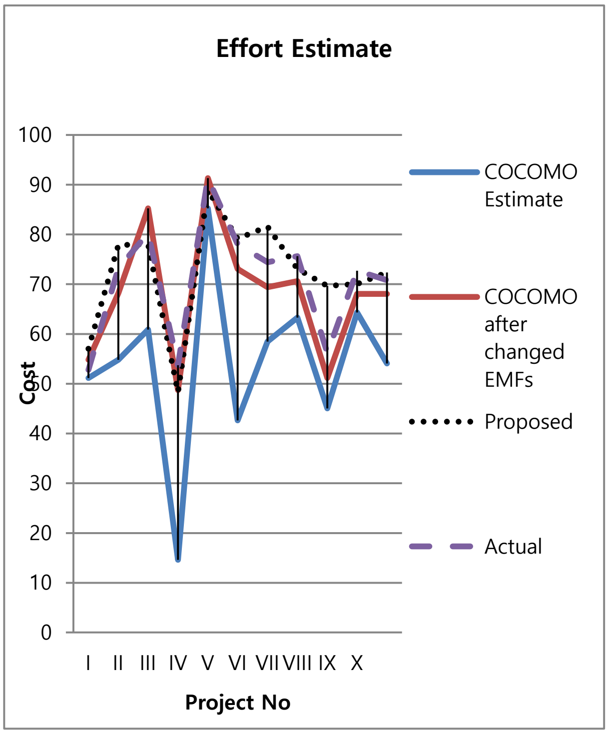
Project No.	Software Platform	Project Size	Effort estimate (in thousands)			
			COCOMO Estimate	COCOMO after changed EMFs	Proposed	Actual
I	ASP.Net	32 KLOC approx	51.1454196	54.79866386	57.01641195	52.79866
II	VB 6.0	33 KLOC approx	54.79866386	68.1938928	77.84713706	73.19389
III	ASP.Net	32 KLOC approx	60.88740429	85.242366	78.06488515	80.24237
IV	VB 6.0	35 KLOC approx	14.61297703	48.70992343	48.27442726	53.70992

V	ASP.Net	36 KLOC approx	85.242366	91.33110643	88.89561026	91.33111
VI	VB 6.0	32 KLOC approx	42.621183	73.06488515	79.15362557	78.06489
VII	VB 6.0	38 KLOC approx	58.45190812	69.41164089	81.58912175	74.41164
VIII	ASP.Net	34 KLOC approx	63.32290046	70.62938897	73.06488515	75.62939
IX	VB 6.0	35 KLOC approx	45.05667917	51.1454196	69.6696562	56.14542
X	ASP.Net	35 KLOC approx	64.32290046	68.05858746	70.06397337	72.72525
		Average	54.0462402	68.05858746	72.36397337	70.82525

$$\textbf{Average Performance Percentage} = \frac{\textit{Actual Cost}}{\textit{Proposed Model Estimate}} * \mathbf{100}$$

$$= \frac{70.82}{72.36} * \mathbf{100}$$

$$= \mathbf{97\%}$$



The average performance is very convincing and matches to a tune of 97% with the actual cost which justifies the accuracy of the model.

6 Conclusion

Harnessing the potential of flexible logic reasoning with multi-valued truth values, the fuzzy inference system is very helpful in deducing the effect of multiple parameters on a decision system. This paper has attempted to superimpose this feature with the proven Algorithmic cost model like COCOMO to reinforce the cost estimation approach with more precise analysis of cause-effect relationship. The proposed approach shows convincing results, matching up to 97%, when compared with the actual cost of the projects. The correctness of the proposed approach lies in the accuracy Rule-base which captures the logic. It is the basis of any inference system and is based on the past experience of project manager and historical data and can be validated which otherwise is also true for the Algorithmic Cost models. The approach would help in finding the trade-offs between various cost drivers so that the cost change is minimal when dealing with dynamic projects. The future work would include justifying the work with other soft-computing techniques.

References

[1] Attarzadeh, I., Siew Hock Ow,”A novel soft computing model to increase the accuracy of software development cost estimation”, Published in 2nd International Conference on Computer and Automation Engineering (ICCAE), 2010, Volume: 3 , Pages: 603 - 607, DOI: 10.1109/ICCAE.2010.5451810

[2] Attarzadeh, I.; Siew Hock Ow, “Improving estimation accuracy of the COCOMO II using an adaptive fuzzy logic model”, IEEE International Conference on Fuzzy Systems (FUZZ), 2011, Pages: 2458 - 2464, DOI: 10.1109/FUZZY.2011.6007471

[3] Rama, S.P., “Analytical structure of a fuzzy logic controller for software development effort estimation”, International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO), Year: 2015 , Pages: 1 - 4, DOI: 10.1109/EESCO.2015.7253928

[4] Attarzadeh, I.; Siew Hock Ow, Proposing a New High Performance Model for Software Cost Estimation””, ICCEE '09. Second International Conference on Computer and Electrical Engineering, 2009, Volume: 2 Pages: 112 - 116, DOI: 10.1109/ICCEE.2009.97

[6] Huang, X.; Ho, D.; Ren, J.; Capretz, L.F. , “A neuro-fuzzy tool for software estimation”, 20th IEEE International Conference on Software Maintenance, 2004. Proceedings. Page: 520, DOI: 10.1109/ICSM.2004.1357862

[7] Mirseidova, S.; Atymtayeva, L., “Definition of software metrics for software project development by using fuzzy sets and logic “,13th International Symposium on Advanced Intelligent Systems (ISIS), Joint 6th International Conference on Soft Computing and Intelligent Systems (SCIS) and 2012, Pages: 272 - 276, DOI: 10.1109/SCIS-ISIS.2012.6505336

[8] Kushwaha, N.; Suryakant, “Software cost estimation using the improved fuzzy logic framework”, Conference on IT in Business, Industry and Government (CSIBIG), 2014, Pages: 1 - 5, DOI: 10.1109/CSIBIG.2014.7056959

[9] [BOE81] Boehm, B., *Software Engineering Economics,* Prentice-Hall, 1981.

[10] [PUT92] Putnam, L. and W. Myers, *Measures for Excellence,* Yourdon Press, 1992.

[11] Ali Bou Nassif,1,2 Mohammad Azzeh,3 Ali Idri,4 and Alain Abran, Software Development Effort Estimation Using Regression Fuzzy Models, Computational Intelligence and Neuroscience, Hindawi publications, Feb 2019, Volume 2019 |Article ID 8367214 | 17 pages | https://doi.org/10.1155/2019/8367214

[12] D.Manikavelan, R. Ponnusamy, Software quality analysis based on cost and error using fuzzy combined COCOMO model, Journal of Ambient Intelligence and Humanized Computing (2020), Springerlink, March 2020

[13] Iman Attarzadeh ; Siew Hock Ow, Proposing a new software cost estimation model based on artificial neural networks, 2nd International Conference on Computer Engineering and Technology, IEEE April 2020, **DOI:** 10.1109/ICCET.2010.5485840

Dr. Shrabani Mallick, is Lecturer in the Dept of Computer Science Engineering at Dr. B. R Ambedkar Institute of Technology, Port Blair Andaman and Nicobar Islands. She received her Ph.D degree from Motilal Nehru National Institute of Technology, Allahabad, Uttar Pradesh India and has 18 years of teaching experience. Her research interest is Distributed Computing, Machine Learning, Internet of Things and Software Engineering.

Dr. Dharmender Singh Kushwaha is Professor in the Dept of Computer Science Engineering at Motilal Nehru National Institute of Technology, Allahabad, Uttar Pradesh India. He has 28 years of teaching experience and is a life member of IEEE and Computer Society of India. His research interest is Distributed System and cloud Computing, Web Services and Data Structures, Internet of Things and Software Engineering.