A NOVEL SOFTWARE INTERVAL TYPE - 2 FUZZY EFFORT ESTIMATION MODEL USING S-FUZZY CONTROLLER WITH MEAN AND STANDARD DEVIATION

Mr. Peram Subba Rao¹, Dr.K.Venkata Rao², Dr.P.Suresh Varma³

¹Astt.Prof. in School of Computing, department of Information Technology, Vignan University, Vadlamudi, Guntur (Dt), Andhra Pradesh, India.
²Prof. and HOD Department of CSE, Vignan’s Institute of Information Technology, Gajuwaka, Visakhapatnam (Dt), Andhra Pradesh, India.
³Dean CDC & Professor, Adikavi Nannaya University, Rajahmundry, W. Godavari (Dt), Andhra Pradesh, India.

ABSTRACT

Now-a-days software cost estimation has prominent role in software engineering practice. Cost estimation of software projects is one of the most desired capabilities in software development organization. It helps the customer to make successful investments and also assist software project manager in upcoming appropriate plans for projects besides making reasonable decision during project execution. In order to fulfill the requirements of cost estimation time effective software is required. By keeping in view of all these requirements, my paper introduces a novel model of interval type-2 fuzzy logic estimation effort in software development. This paper touches upon MATLAB which is tuning parameters for various cost estimation models. Published software projects data model performance is used along with comparison of my novel model and existing ubiquitous models.

Keywords: Cost estimation, fuzzy logic, Interval type-2, membership function, regression analysis.
1. INTRODUCTION

Software cost estimation has vital role in software engineering practice. In the last three decades, many software estimation models and methods have been proposed and used, such as COCOMO, SLIM, SEER-SEM, Price-S, FP, Delphi, Halsted Equation, Bailey-Basili, Doty, and Anish Mittal Model and etc COCOMO [2, 4, 5, 13-18]. It determines the success or failure of contract negotiations and project execution. Cost estimation’s deliverables are effort, schedule, and staff requirements. These are valuable pieces of information for project development and implementation. They are used by key inputs for project bidding and proposals, budget and staff allocation, project planning, progress monitoring and control.

If the estimates are not reasonable and unreliable, there is possibility of project failure. For example, it is evident in CompTIA survey of 1,000 IT respondents in the year 2007. Identifying these failures which lead to failure of IT projects. This failure is only due to unrealistic resource estimation.

So, recognizing the importance of software estimation, the software engineer’s community has put tremendous effort to develop models especially to help estimators generate accurate cost for software projects.

Software engineering cost (and schedule) models and estimation techniques are used for other purposes. Which includes budgeting, tradeoff and risk analysis, project planning & control and software improvement investment analysis. Significant research on software cost modeling began with an extensive 1965 SDC study of the 104 attributes of 169 software projects. This led to some useful partial models in the late 1960s and early 1970s. The late 1970s produced a flowering of more robust models such as SLIM, Checkpoint, PRICE-S, SEER and COCOMO [2, 4, 5, 13-18]. Although most of these researchers started working on developing models of cost estimation at the same time, all of them have faced the same problem as such as if software grow in size and importance it also grows in complexity, makes it very difficult to predict accurate cost of software development. These dynamic fields of software estimation sustain the interests of researchers who have been succeeded in setting the stepping-stones of software engineering cost models.

As every coin has two sides, this field of software engineering cost models has its own pitfalls. Due to fast changing nature of software development made it very difficult to develop parametric models that yield high accuracy for software development in all domains. Software development costs continue to increase and practitioners may express their concern and inability to predict accurate cost. One of the most important objectives of the software engineering community is the development of useful models that constructively explains the development life-cycle. Besides predicting the cost for developing a software product. Many software estimation models have evolved in the last two decades based on the pioneering efforts of the researchers.

A recent survey shows that the software projects overrun the cost estimation, with several other problems are found in this area like unrealistic over-optimum, complexity, and overlooked tasks [1, 15]. So, to overcome all these problems in software development, new models approached which researchers had showed attention in 1990’s. They are artificial neural networks, fuzzy logic models and genetic algorithms [13-14]. Among these models, it is proved that fuzzy logic is the best powerful linguistic representation with exact inputs and outputs. It is also based on model building with logic concepts introduced by Lofti A. Zadeh [3].
1.1 Membership Functions

Fuzzy numbers are one of the ways to describe data vagueness, obscurity and imprecision. A fuzzy number is an extension of a regular number in the sense that it does not refer to one single value but rather to a connected set of possible values, where each possible value has its own weight between ‘0’ and ‘1’. This weight is called the membership function. The membership function is increasing towards the mean and decreasing away from it. The Fuzzy number can be of three types 1) Triangular fuzzy Number 2) Trapezoidal fuzzy number 3) Bell shaped fuzzy number. Figure 1 shows the three curves [3]

![Figure 1: Types of Membership Functions](image)

Fuzziness in a fuzzy set is characterized by its membership functions. A membership function (MF) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. The graphical representations include different shapes. There are certain restrictions regarding the shapes used. The “shape” of the membership function is an important criterion that has to be considered. There are different methods to form membership functions.

Zadeh proposed [3] a series of membership functions that could be classified into two groups: those made up of straight lines, or “linear,” and Gaussian forms, or “curved.” [3] Based on this criterion the membership function can be one of the following types. [3]

1.2 S Function

Defined by its lower limit a, its upper limit b, and the value m or point of inflection so that a < m < b. A typical value is: m = (a + b) / 2. Growth is slower when the distance a – b increases.

\[
S(x) = \begin{cases} 
0 & \text{if } x \leq a \\
\frac{1}{2} \left( \frac{x-a}{b-a} \right)^2 & \text{if } x \in (a, m) \\
1 - \frac{1}{2} \left( \frac{x-a}{b-a} \right)^2 & \text{if } x \in (m, b) \\
1 & \text{if } x \geq b 
\end{cases} \quad \ldots (1)
\]

![Figure 2: S Fuzzy Set](image)
2. EFFORT ESTIMATION MODELS LITERATURE REVIEW

Within last few decades, to improve the accuracy of cost estimation many software cost estimation models [2], [4-12] were introduced. It seems to be impractical to expect very accurate cost estimates of any software because of the inherent obscurity in software development projects and the complex and dynamic interaction of factors that impact software development cost use. Still, it is likely that estimates can be improved because software development cost estimates systematically overoptimistic and very inconsistent. The primary objective of the software engineers is to develop required models by using software cost which can be accurately estimated. Estimation models use KDLOC (Thousands of Delivered Lines of Code) as the primary input. This input is not sufficient for estimating the accurate cost of products. Several other parameters have to be considered.

COCOMO81 Model
Boehm described COCOMO as a collection of three variants: basic model, intermediate model, detailed model [12].

Basic model
\[ \text{Effort} = a \times \text{size}^b \quad \ldots (2) \]

Intermediate model
\[ \text{Effort} = (a \times \text{size}^b) \times \text{EAF} \quad \ldots (3) \]

Detailed model
\[ \text{Effort} = (a \times \text{size}^b) \times \text{EAF} \times \text{sum}(W_i) \quad \ldots (4) \]

COCOMO II Model
Boehm and his colleagues have refined and updated COCOMO called as COCOMO II. This consists of application composition model, early design model, post architecture model.

The Early Design Model
\[ \text{Effort} = a \times \text{KLOC} \times \text{EAF} \quad \ldots (5) \]

The Post Architecture Model
\[ \text{Effort} = (a \times \text{size}^b) \times \text{EAF} \quad \ldots (6) \]

Doty Model
\[ \text{Effort} = 5.288(\text{KLOC})^{1.047} \quad \ldots (7) \]

Halsted Equation Model
\[ \text{Effort} = 5.2 (\text{KLOC})^{1.50} \quad \ldots (8) \]

Bailey-Basili Model
\[ \text{Effort} = 5.5 + 0.73 (\text{KLOC})^{1.16} \quad \ldots (9) \]

Mittal Model
Fuzzification:
\[ u(E) = \begin{cases} 
0 & \text{if } E \leq a \alpha^2 \\
\frac{E - a \alpha^2}{m - a \alpha^2} & \text{if } a \alpha^2 \leq E \leq u \beta^2 \\
\frac{E - u \beta^2}{\beta - u \beta^2} & \text{if } u \beta^2 \leq E \leq a \beta^2 \\
1 & \text{if } E \geq a \beta^2
\end{cases} \quad \text{(10)}

Defuzzification:
\[ E = \frac{w_1(aw^2) + w_2(am^2) + w_3(a \beta^2)}{(w_1 + w_2 + w_3)} \quad \text{(11)}

Harish model 1
\[ \text{Effort} (E) = \frac{w_1(aw^2) + w_2(am^2) + w_3(a \beta^2)}{w_1 + w_2 + w_3} + \varphi (E) + d \quad \text{(12)}

Harish model 2
\[ \text{Effort} (E) = \frac{w_1(a \alpha^2) + w_2(a m^2) + w_3(a \beta^2)}{w_1 + w_2 + w_3} + \varphi (E) + d \quad \text{(13)}

3. PROPOSED METHOD

Introduction on “interval type-2 fuzzy logic” for software cost estimation.

Structure of Type-2 fuzzy logic [20-27] is shown above. Fuzzification is the process of which it translates inputs (real values) to fuzzy values. Inference System applies a fuzzy reasoning mechanism to obtain a fuzzy output. Knowledge Base contains a set of fuzzy rules, it is of the form \( R^i : \text{if } x_1 \text{ is } F_1^i \text{ and } \ldots \text{and } x_n \text{ is } F_n^i \text{ then } Y \text{ is } G^i \text{, i=1,2,} \ldots \text{m and a membership functions set known as the database. Type Reducer transforms a Fuzzy Set into a Type-1 Fuzzy Set. The defuzzification traduces one output to precise values.}

Interval Type-2 Fuzzy Logic
A type-2 fuzzy set [20-27], denoted as \( A \), is characterized by a type-2 Membership Function (MF), \( \mu_A(x, u) \), where \( x \in X \) and \( u \in J_x \), i.e.
\[ \tilde{A} = \{(x, u), \mu_M(x, u)) \mid \forall x \in X, \forall u \in J_x \subseteq [0,1] \} \quad \text{(14)} \]
In which \( 0 \leq \mu_A(x, u) \leq 1 \), if the universes of discourse X and the domain of secondary membership function \( J_x \) are continuous, \( A \) can be expressed as:

\[
A = \int_{x \in X} \int_{u \in J_x} \mu_A(x, \mu) / (x, \mu), \quad j_x \subseteq [0, 1] \quad \ldots (15)
\]

Where \( \int \) denotes union over all admissible \( x \) and \( \mu \). If the universes of discourse \( X \) and \( J_x \) are both discrete, \( \int \) is replaced by \( \Sigma \), \( A \) can also be expressed as:

\[
A = \sum_{i=1}^N \sum_{j=1}^M \mu_A(x_i, \mu_j) / (x_i, \mu_j) \quad \ldots (16)
\]

If universes of discourse \( X \) and \( J_x \) are both discrete, (4) can be expressed as:

\[
\tilde{A} = \sum_{x \in X} \sum_{\mu \in J_x} \mu_A(x, \mu) = \left[ \sum_{x \in X} \sum_{\mu \in J_x} \mu_A(x, \mu) \right] / x \quad \ldots (19)
\]

In the above equation “+” denotes union. Uncertainty in the primary memberships of an IT2 FS consists of a bounded region named as Footprint of Uncertainty (FOU) [20-27]. It is the union of all primary memberships, i.e.

\[
FOU(A) = U_{x \in X} J_x \quad \ldots (17)
\]

This is a vertical-slice representation of FOU, because each of primary membership is a vertical slice. The Upper Membership Function (UMF) and Lower Membership Function (LMF) of \( A \) are two T1 MFs that bound the FOU. The UMF is associated with the upper bound of FOU \( (A) \) and is denoted as \( \mu_A(x) \), \( \forall x \in X \) and the LMF is associated with the lower bound of FOU \( (A) \) and is denoted as \( \mu_A(x) \), \( \forall x \in X \), i.e.

\[
\mu_A(x) = \frac{FOU(A)}{X} \quad \forall x \in X \quad \ldots (18)
\]

For an IT2FS, \( J_x = [\mu_A(x), \mu_A(x)], \forall x \in x \) Therefore, the IT2FS \( A \) can be denoted as.

\[
\sum_{x \in X} [\mu_A(x), \mu_A(x)] / x \quad (in \ discrete \ situation) \quad or \quad \ldots (19)
\]

\[
\text{Effort} = \int \left[ \mu_A(x), \mu_A(x) \right] / x \quad (in \ continuous \ situation) \quad \ldots (20)
\]

The Upper Membership Function (UMF) and Lower Membership Function (LMF) of \( A \) are two T1 MFs that bound the FOU. For an interval type-2 fuzzy system (ITF2S).
\[
J_{px} = \frac{\mu_{PL} + \mu_{PD} + \mu_{PR}}{2}, \quad J_{Nx} = \frac{\mu_{NL} + \mu_{ND} + \mu_{NR}}{2}
\]
\[
\begin{bmatrix}
\mu_{PL}(x), \mu_{PR}(x) \\
\mu_{NL}(x), \mu_{NR}(x)
\end{bmatrix}
\]  
(21)
(22)

Are the firing intervals for positive and negative functions \(\mu_{PL}, \mu_{PR}\) are left hand side uncertainty region. \(\mu_{PL}, \mu_{PR}\) are right hand side uncertainty region. Similarly for negative membership function.

### 3.1 Model Description

In this model I consider the mean of FOU `s as a firing interval in interval type-2 [20-27], to estimate the cost (Effort) of the software.

1. In the first step by regression analysis (power regression) estimate the parameters of a,b (the amplitude a and the exponent b).
2. In the second step, the variable “size” is fuzzified by two input fuzzy sets named “Positive” and “Negative” respective. The mean of the size(L) is input for determining the fuzzy memberships.

![Figure 4: Universe Of Discourse](image)

The membership value \(\mu_p(x_i)\) and \(\mu_n(x_i)\) is either 0 or 1 when \(x_i\) is outside the interval [-L,L]. This process is known as fuzzification.

3. Next step is to apply S membership function to reduce the uncertainty (FOU). We calculate the uncertainty at the left side (LMF) and right side (UMF). This is known as fuzzy inference.

4. After identifying the LMF and UMF for the both positive and negative membership functions, we the mean of the LMF and UMF FOU as a firing interval for converting Type-2 into Type-1 fuzzy sets. This is known as Type Reducer.

5. Finally, In order to convert Fuzzy values into output (Effort), weighted average defuzzification (centroid) method is used.
3.2 Model Analysis: Regression Analysis

By using polynomial regression, \([www.xuru.com]\) a,b parameters \(Y = ax^b\) is calculated. Where x is the variable along the x-axis. The function is based on linear regression with both axis are scaled logarithmically.

3.3 S membership function(SMF)

We use S function Fuzzy number S(N) which is defined as follows:

\[
S(n) = \begin{cases} 
0 & \text{if } n \leq \alpha \\
\frac{2(n-\alpha)}{\beta-\alpha} & \text{if } \alpha < n < \beta \\
1 - \frac{2(n-\beta)}{\beta-\alpha} & \text{if } \beta \leq n
\end{cases} \quad \ldots (23)
\]

**Figure 5:** Representation of Effort

Where \(n\) is size as input, \(E\) is effort as output and \(\alpha, \beta, m\) are the parameters of membership function \(S(n)\).

Fuzziness of SFN \((\alpha, m, \beta)\) is defined as:

\[
m = \text{model value} \quad \alpha = \text{Left Boundary}, \beta = \text{is right boundary}
\]

Fuzziness of TFN \((F)\) = \(\frac{\beta - \alpha}{2m}\), \(0 < F < 1 \quad \ldots (24)\)

The higher the value of fuzziness, the more fuzzy in SFN. The value of fuzziness to be taken depends upon the confidence of the estimator. A confident estimator can take smaller values of \(F\). Let \((m, 0)\) divides internally, the base of the triangle in ration \(K : 1\) why \(K\) in the real positive number.

So that \(m = \frac{a + \frac{K \beta}{K+1}}{K+1} \quad \ldots (25)\)

As per the above definitions. \(F = \frac{\beta - \alpha}{2m} \quad \ldots (26)\)
So \[ \alpha = \left(1 - \frac{2F}{K + 1}\right)^m \quad \text{and} \quad \beta = \left(1 + \frac{2F}{K + 1}\right)^m \quad \ldots (27) \]

### 3.4 Interval Type-2 TMF \((a, m, \beta)\) Firing intervals

\[
\begin{align*}
J_{P_X} &= \left[\frac{\mu_{PL1} + \mu_{PL2}}{2}, \frac{\mu_{PR1} + \mu_{PR2}}{2}\right] = \left[\bar{\mu}_P(x_i), \tilde{\mu}_P(x_i)\right] \\
J_{N_X} &= \left[\frac{\mu_{NL1} + \mu_{NL2}}{2}, \frac{\mu_{NR1} + \mu_{NR2}}{2}\right] = \left[\bar{\mu}_N(x_i), \tilde{\mu}_N(x_i)\right]
\end{align*}
\]

### 3.5 Defuzzification

In this model, centroid method (weights average) is considered, which is of the form

\[
C = \frac{\sum_{i=1}^{N} w_i \mu_i}{\sum_{i=1}^{N} w_i} \quad \ldots (28)
\]

### 4. EXPERIMENTAL RESULTS AND DISCUSSION

For the membership functions the L value is the mean of the input sizes [5, 19] i.e. 207.3385 and stddev() is 186.3325 (L1=393.671, L2= 21.006). By applying polynomial regression [www.xuru.org] analysis for the input sizes and effort is obtained a=70.737 and b=0.004.

By applying S membership function for the membership functions the left and right boundaries are \([\mu_P(\alpha, m, \beta), \mu_N(\alpha, m, \beta)]\) measured with \(\alpha=0.9m\) and \(\beta=1.1m\).

Footprint of uncertainty intervals for the \(\mu_P\) is [0.501212 to 0.612593] for left hand side i.e LMF and [0.9 to 1.1] for right hand side i.e UMF. Footprint of uncertainty intervals for the \(\mu_N\) is [0.13737 to 0.167896] for left hand side i.e LMF and [0.206148 to 0.251959] for right hand side i.e UMF. The means of FOU intervals is taken as firing strength.

\[
\begin{align*}
JP_X &= [\mu_P(X_i), \mu_P(X_i)] = [0.556903, 1] \\
JN_X &= [\mu_N(X_i), \mu_N(X_i)] = [0.152633, 0.229054]
\end{align*}
\]

The type reducer action by using the S membership function and defuzzification is done through centroid method and results are shown in the table II.
4.1 Estimated Efforts in Man-Months of Various Models

### TABLE I

**EFFORTS OF VARIOUS MODELS**

| ProjNo | KLOC | Actual Effort | COCOMO Basic | Detailed | Envy | Design | Post Arch | Effort Model | Halstead Eq | Bailey-Basili | Harlan Model | Proposed Model |
|--------|------|---------------|--------------|----------|------|--------|----------|--------------|-------------|--------------|--------------|--------------|---------------|
| 1      | 39   | 72            | 292.13       | 227.21   | 24.4 | 107.7  | 211.9    | 245          | 1266.856    | 55.65212    | 59.56694    | 126.78828    |
| 2      | 40.5 | 82.3         | 305.67       | 231.74   | 25.8 | 111.8  | 221.3    | 254.9        | 1340.2902   | 59.91072    | 61.60873    | 126.51629    |
| 3      | 50   | 94           | 393.81       | 308.14   | 33.8 | 131.1  | 282      | 317.8        | 1388.4776   | 73.75993    | 70.28889    | 124.80212    |
| 4      | 128.6| 230.7        | 1222.9       | 953.15   | 357.2 | 351.3  | 835.7    | 854.4        | 7583.4145   | 209.69732   | 173.07023   | 117.29613    |
| 5      | 163.4| 157         | 1605.16      | 1249.25  | 1349.2 | 447.6  | 1085.3   | 1083.8       | 10962.49    | 271.25092   | 153.18057   | 117.12583    |
| 6      | 164.5| 246.9        | 1654.84      | 1750.87  | 1533.3 | 455.0  | 1111.6   | 1107.8       | 11091.01    | 277.75888   | 155.83186   | 117.21940    |
| 7      | 200  | 207.49       | 1815.82      | 1745.1   | 552.2 | 1288.8 | 1567.6   | 14707.821   | 346.31418   | 176.6537    | 115.34193    |
| 8      | 214.4| 225.12       | 2175.42      | 2196.9   | 592   | 1508.4 | 1589.1   | 15724.285   | 374.93974   | 185.01721   | 121.81589    |
| 9      | 253.6| 267         | 2762.57      | 2318.21  | 2250.4 | 700.2  | 1828.3   | 1739.5       | 21008.382   | 453.33959   | 206.87425   | 126.60496    |
| 10     | 254.2| 257.8       | 2779.21      | 2154.61  | 2357  | 701.9  | 1829.8   | 1743.9       | 21074.955   | 457.61312   | 207.19998   | 126.71344    |
| 11     | 289  | 316         | 3233.12      | 2513.25  | 2714.3 | 798    | 2130.7   | 1994.6       | 25547.6    | 527.83333   | 225.05449   | 134.06897    |
| 12     | 449.9| 336.3       | 5457.94      | 4727.62  | 6016.0 | 1222.0 | 3728.9   | 3170.3       | 19922.851   | 783.34577   | 303.89744   | 142.14851    |
| 13     | 450  | 1107.31     | 5497.4       | 4275.76  | 6417.8 | 1225.2 | 3528.0   | 3171.1       | 19459.884   | 785.57043   | 302.93423   | 144.79948    |

### TABLE II

**EFFORT ESTIMATION**

<table>
<thead>
<tr>
<th>Sno</th>
<th>Site</th>
<th>Actual Effort</th>
<th>Effort Estimation</th>
<th>Effort Model</th>
<th>Halstead Eq</th>
<th>Bailey-Basili</th>
<th>Harlan Model</th>
<th>Proposed Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>39</td>
<td>21.75</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
</tr>
<tr>
<td>2</td>
<td>40.5</td>
<td>22.04</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
</tr>
<tr>
<td>3</td>
<td>50</td>
<td>50.86</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
</tr>
<tr>
<td>4</td>
<td>128.6</td>
<td>128.6</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
</tr>
<tr>
<td>5</td>
<td>163.4</td>
<td>163.4</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
</tr>
<tr>
<td>6</td>
<td>164.5</td>
<td>164.5</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
</tr>
<tr>
<td>7</td>
<td>200</td>
<td>200</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
</tr>
<tr>
<td>8</td>
<td>214.4</td>
<td>214.4</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
</tr>
<tr>
<td>9</td>
<td>253.6</td>
<td>253.6</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
</tr>
<tr>
<td>10</td>
<td>254.2</td>
<td>254.2</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
</tr>
<tr>
<td>11</td>
<td>289</td>
<td>289</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
</tr>
<tr>
<td>12</td>
<td>449.9</td>
<td>449.9</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
</tr>
<tr>
<td>13</td>
<td>450</td>
<td>450</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
</tr>
</tbody>
</table>

4.2 Following graph comparison of interval type-2 proposed model results with measured efforts

![Figure 6: Measured Effort Vs Proposed Model Estimated Effort](image)
4.3 Performance Measures

The performance of proposed software effort estimation model is assessed by comparing against various software cost estimation models. The methodology used in empirical evaluation is described as follows:

- For each model, using MRE we evaluate the impact of estimation accuracy using (MARE, VARE, StdDev) evaluation criteria.
- Criterion for measurement of software effort estimation model performance. The three criterions are

1. **Mean Absolute Relative Error (MARE)**

   $$\% \text{ MARE} = \frac{\text{Mean} \left[ \frac{\text{abs} (\text{Measured Effort} - \text{Estimated Effort})}{\text{Measured Effort}} \right] \times 100}{\text{Actual Effort}}$$

2. **Variance Absolute Relative Error (VARE)**

   $$\% \text{ VARE} = \frac{\text{Var} \left[ \frac{\text{abs} (\text{Measured Effort} - \text{Estimated Effort})}{\text{Measured Effort}} \right] \times 100}{\text{Actual Effort}}$$

3. **Standard Deviation (StdDev)**

   $$\% \text{ StdDev} = \sqrt{\text{VARE} \%}$$
4.4 Comparison of various models

<table>
<thead>
<tr>
<th>Model Type</th>
<th>MARE %</th>
<th>VARE %</th>
<th>StdDev</th>
</tr>
</thead>
<tbody>
<tr>
<td>COCOMO Basic Model</td>
<td>13307.66</td>
<td>609614.2</td>
<td>780.7796</td>
</tr>
<tr>
<td>COCOMO Inter(Nova)</td>
<td>10061.5</td>
<td>368779.3</td>
<td>607.27198</td>
</tr>
<tr>
<td>Detailed (Nov)</td>
<td>10970.6</td>
<td>430138.1</td>
<td>655.84913</td>
</tr>
<tr>
<td>Early Design Model (High)</td>
<td>2561.3</td>
<td>35238.3</td>
<td>187.7187</td>
</tr>
<tr>
<td>Post Arch Model (H - H)</td>
<td>8438</td>
<td>259268.1</td>
<td>509.1838</td>
</tr>
<tr>
<td>Dory Model</td>
<td>8186.1</td>
<td>223081.7</td>
<td>472.8152</td>
</tr>
<tr>
<td>Halsted Eq</td>
<td>105605.2</td>
<td>40885219</td>
<td>6992.591</td>
</tr>
<tr>
<td>Bailey-Basili</td>
<td>1325</td>
<td>13147.42</td>
<td>114.6622</td>
</tr>
<tr>
<td>Proposed Model</td>
<td>599.6442</td>
<td>407.2915</td>
<td>20.18146</td>
</tr>
</tbody>
</table>

**Figure 8:** Mean Absolute Relative Error for above models

**Figure 9:** Variance Absolute Relative Error for above models
5. CONCLUSION & FUTURE WORK

Software development life cycle is an important for project managers to estimate the accuracy and reliability at the early stages of software development. This paper postulates about the fuzzy software cost estimation model and with other popular software cost estimation models. It concludes by empirical evaluation is better software effort with proposed and traditional estimation models by MARE, VARE and Standard Deviation evaluation criteria. To identify the problem of obscurity and vagueness that are existed in software effort drivers’ fuzzy logic methods are applied. This proves the fuzzy logic application is used in software engineering successfully and the work of Interval Type-2 fuzzy sets can be applied to other models of software cost estimation.

REFERENCES

[10] Kim Johnson, 'Software Cost Estimation: Metrics and Models,' Department of Computer Science University of Calgary,


[22] Jerry M. Mendel, Jerry M. Mendel, Robert I. John]” Standard Background Material AboutInterval Type-2 Fuzzy Logic Systems That Can Be Used By All Authors”.


[27] Juan R. Castro, Oscar Castillo, Luis G. Martínez” Interval Type-2 Fuzzy Logic Toolbox”.2007
