Neutrosophic Logic for Software Effort Estimation: An Improvement over Fuzzy Logic

Arun Sharma, Pankaj Goel, Prachi Singh

Abstract: Software effort estimation is big and active research area. Software effort estimation is useful for time and efforts required to perform a particular task. But, it is very rare to estimate the effort with high level of reliability. There are various approaches to estimate the software application effort. In the present paper, to estimate the effort for software applications efforts, neutrosophic logic approach is used. Neutrosophic logic is a mathematical model for ambiguity, uncertainty, incompleteness, vagueness, redundancy and contradiction and inconsistency in data. It is the extension to the fuzzy logic. It is capable of handling those errors which are not handled by fuzzy logic like indeterminacy in the data. Neutrosophic logic gives the results very similar to human thinking. The present work concludes that neutrosophic logic optimizes the performance of fuzzy logic while calculating the software efforts.

Keywords: Software Effort Estimation, Fuzzy Logic, Neutrosophic logic.

I. INTRODUCTION

Precise software effort estimation is considered as a challenging task in process of software development. Accuracy is an important factor for efficient cost estimation. If there is deviation between approximate and accurate value of effort than it can be dangerous for process of project. Both low effort estimation and high effort estimation causes trouble in life cycle of project. Due to this trouble, project may take longer time in process and greater effort than others.

It is important to estimate the accurate cost because it helps in classifying and prioritizing the project development. It also helps to determine what and how well these resources used to commit to the project [1]. There are various approaches used to estimate the effort. In previous years, effort estimation was done with the help of the experienced project manager, which seems to be very difficult for every project.

Cost is increased exponentially because this approach was not reliable and repeatable. It is useful in case if project is relevant to previous projects.

For better planning, it requires the effort estimation at the early stage of software life cycle, which is not an easy task.

According to Sommerville [2], there are various factors which make it difficult, like high user requirements and software run on new technology etc. According to Macconnell [3], software development is done gradually. At initial stage, a very rough idea of the software is made, estimate of time and efforts are also build in vaguely. When software comes into a clear picture then estimation will also become clear. At the early stage of estimation, due to the uncertainty, it is difficult to estimate the projects at early stages. [4]

To handle the uncertainty, extended version of fuzzy logic is used which is also known as neutrosophic logic. It is a mathematical model of uncertainty, incompleteness, imprecise, ambiguous. In this new classification technique is used to estimate the effort with level of uncertainty i.e. neutrosophic classifier. possible.

II. SOFT COMPUTING

Soft Computing is a problem solving technique which involves reasoning, thinking, deduction, grouping, classification of different quantities. The purpose of soft computing is to overcome the problem encountered with the problems of the real world. It does not require any definition and maths for system component. Soft computing aims to achieve robust and low cost solutions. The advantage of using soft computing is that it is able to compute those problems which cannot be clarified. With the help of soft computing, an autonomous and automated intelligence system can be designed.

Fuzzy sets introduced by Prof. Zadeh [5] allows to introduce degree of membership. It is used to solve complex problems which are difficult to be solved with traditional mathematical methods. It is a tool widely used in number of applications has proved to be less complex and increased accessibility of control theory [6].

III. NEUTROSOPHIC LOGIC

Neutrosophic logic [7] depicts the mathematical modeling of handling uncertainty [8]. Different propositions of neutrosophic logic are estimated with the % of truth T, indeterminacy I and falsity F in the subsets represented, where T, I, F are real subsets of [0,1] [9]. These may be discrete or continuous; single-element, overlap, finite, or infinite [10]. It is useful in various cases where percentage of truth and falsity cannot be measured exactly. The value of membership function is varied from 0 to 1. In definite environment, the knowledge received by human from external world is imperfect, so it is difficult to conclude to decision.
Neutrosophic Logic for Software Effort Estimation: An Improvement over Fuzzy Logic

But neutrosophic logic results are very similar to what human being thinks. For example, “Person X is tall”, in this situation, human cannot assess a precise decision in True or False or no form. There is a indeterminacy which is the factor of not knowing the value, between truth and falsehood. In this type of situation neutrosophic logic is best suited similar to human brain reasoning.[11] The value of the preposition “Person X is tall” may be differ from person to person and place to place. Neutrosophy also allows change in values and depends on person to person [10]. For example: proposition “Person X is tall” may yield (T=0.50, I=0.20, F=0.35) as a neutrosophic component if observed by one person and it is different (T=0.45, I=0.25, F=0.60) for a different person.

IV. NEUTROSOFPHIC LOGIC VERSUS FUZZY LOGIC

Neutrosophic logic is better than fuzzy logic because of various reasons. These reasons are discussed as follows:

- Fuzzy logic argues that the sum of membership and non-membership value of an element to a particular class should be one[12-15]
- Fuzzy logic cannot deal with the indeterminacy due to incomplete knowledge and observation error (errors while acquiring information).
- In case of fuzzy logic, non membership value is the complement of membership value, both ranging in[0 1]. However, in neutrosophic logic, absolute true/false values and relative true/false values are totally different and the values of T, I and F are not necessarily lying in the range of zero to one [10].

V. RELATED WORK

This paper describes a neutrosophic classifier for software effort estimation. It used neutrosophic logic to perform its working. Neutrosophic logic allows to handle acquisition errors efficiently, indeterminacy and stochasticity that cannot handled by the fuzzy logic [11]. According to Batra et.al.[16], Software effort estimation is a recent activity in software management and it is a critical issue to accurately and reliably estimate for project manager. In this fuzzy logic approach is used to handle the imprecise data. It is the improved approach of classical intervals in COCOMO models. The fuzzy sets are represented using Gaussian-shaped membership functions which are nearer to actual estimation.

According to Nisar et.al.[1], the overview of various type of collection of software effort estimation techniques is discussed used by fuzzy logic. In software engineering, there are variety of Computational Intelligence techniques but these are not enough mature to estimate. Strengths and weaknesses are also discussed in existing software estimation techniques in industry and literature. This paper proves that fuzzy logic may be an efficient tool to calculate the software effort. McConnell [3] defines that the common story of estimating the software is that development of software is a process of continuous processing. To start a project, there is a diffuse image of roughly idea of project, thus making effort estimation and time required to complete as fuzzy. In this estimation approach, the image of software is also become clear. Therefore, the statement/technique which is used to estimate software project is also a process of gradually continuous process. According to Liming Wu [17], generate/produce good management decisions for accurate software development cost estimation is very critical. It is also very difficult to accurately identify how much time and effort is needed for project managers. There are many ways of estimating the cost of software available by including algorithms for effort estimation, estimates by expert judgment method, analogy, bottom up method, and top down method. To understand which approach is better, knowledge of strengths and weaknesses are very important. Various estimation methods should depends on the environment of the project. According to expert judgement COCOMO2.0 is better than other approaches. According to Kjetil Moløkken et.al.[18], this paper defines the various rules for effort estimation. Survey is done to define the rules, these rules can be:

- Expert effort estimation is the most frequent method which is in used.
- There is no proving for that if formal estimation methods are used than it leads to more accurate estimates.
- It is state that accurate estimation of effort is still a problem. Ansari et al. [19] introduced a new neutrosophic classifier which is the extension to fuzzy classifier. UCI repository dataset is used to implement neutrosophic logic. The result is generated in three components truthness, indeterminacy and falseness. Ansari et al.,[15] in another paper discussed the difference between neutrosophic logic and other logics. Neutrosophic controller is proposed in this paper. It is a special type of system which is more accurate and indeterminacy tolerance than fuzzy logic in working. In such situation where the possibility of incompleteness and indeterminacy of acquired data, fuzzy logic does not work. For such situation neutrosophic logics are developed or introduced. According to Smarandache [20], neutrosophic logic is the theory of everything. Neutrosophic is a three valued infinite logic but if indeterminacy component can be divided or split into one or more component according to each application for example vagueness, ambiguity, unknown, unpredicted, error, etc. this statement stated that neutrosophic is a n-finite logic. Ming Zhang et al. [21] used neutrosophic logic for proposing a new watershed image segmentation approach. Firstly, a given image is mapped with three different subset domains T, I and F. Obtain the binary image than apply the watershed algorithm to get the segmentation result. This method showed better performance on noisy and non-uniform images than other watershed approaches. Athar Kharal [22], proposed a new approach for multi criteria decision making by using neutrosophic logic. This technique allows the decision makers to assign and adjust the degree of satisfaction, not satisfaction and indeterminacy from the criteria. This makes the decision making process or calculation simpler and flexible. According to Smarandache [23], neutrosophic probability can be applied in quantum physics in order to model Heisenberg’s Uncertainty Principle of a particle’s behaviour. Heisenberg's uncertainty principle says that it is impossible fully predicting the behaviour of a particle, also the principle of causality cannot be applied at the atomic level.
The possibility of neutrosophic is useful in those events that are related to some degree of Uncertainty (unknown). According to Juan-juan et al.[24], it is difficult to precisely express the preferences for decision maker while solving MCDM problems if the information is uncertain, imprecise, and incomplete. It is difficult to define the membership degree with one specific value so, it is hard to solve with fuzzy logic. This problem can be handled with neutrosophic because it uses the non-standard interval for each variable.

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections A-D below for more information on proofreading, spelling and grammar.

VI. IMPLEMENTATION

In this proposed approach, neutrosophic logic is used for effort estimation of software in person-hours. Neutrosophic logic systems are similar to their fuzzy logic correspondent parts.

Neutrosophic logic classifier can be implemented with the help of MATLAB. It is recommended that suggested fuzzy logic uses some line of code and gives a defuzzified value, but in neutrosophic classifier, output value should be in specified format such as output (truthness, indeterminacy, falsity). Where truthness T, indeterminacy I and falsity F are the neutrosophic components and these are not dependent of each other. Three fuzzy inference systems (FIS) are designed; T, I and F component with the help of fuzzy logic toolbox in MATLAB. Working of all these three components are not dependent on each other but to capture the T, I and F of both input and output, a interrelationship can be showed or maintained between membership functions of T, I and F of neutrosophic components [11].

Dataset description: dataset for effort estimation is taken from promise repository. UCP (use case point) benchmark dataset is selected from promise repository of estimating efforts of software applications. In this dataset, total number of attributes is 9. Before apply neutrosophic to dataset pre-processing is done. At pre-processing stage, unnecessary attributes are eliminated and combined some attributes to make dataset small for processing. After pre-processing, normalization is done to make the value of every attribute in between 0 and 1.

A. Implementation of Fuzzy Classifier

Fuzzy classifier, Mamdani is used to design the proposed work for implementation using MATLABR2008b for UCP data set. Overlapped membership function has been designed for all attributes.

The membership function is designed using input attributes Actors, UC (use case), CF (complexity factor), Application type and output attribute real_effort_person_hours in Mamdani type fuzzy classifier from UCP data set.

It can be stated that after defuzzification, the generated outputs can be of two types using FIS. If the output is positioned in one of the output class, then it is clear that belongingness of that output is related to a particular class. However, if the output value belongs to the overlapping ranges of classes, then this state gives a clear indication to certain degree of indeterminacy. So that, there is some degree of indeterminacy associated with all overlapping membership functions. [11]

Figures 1a, 2a, 3a, 4a are inputs to the mamdani fuzzy classifier. Fig 1a shows membership function for Actors. Fig 2a shows membership function for UC (use case). Fig 3a shows membership function for CF (complexity factor). Fig 4a shows membership function for Application type.
Neutrosophic Logic for Software Effort Estimation: An Improvement over Fuzzy Logic

Fig 5a

Fig 5a show the membership function for output variable i.e. Real effort in person hours.

After calculating all estimated efforts using fuzzy classifier, compare it with the original efforts and calculate the root mean square error for fuzzy classifier.

In UCP data set, calculated RMSE is 0.253.

B. MATLAB implementation of Neutrosophic classifier

The stages for implementation of neutrosophic logic classification inference is given as follows:

1. Design the sets for each given class by using the given input dataset.

2. The three inference systems i.e. Truth component, Indeterminacy and Falsity are developed using FIS editor which are not dependent of each other:

3. Design the inference system for truth component by using the available sets, the steps for implementation of truth component are as under:

   a. By using membership editor function, Membership functions are designed for all input and output variables, so that there is no overlapped area between any of two membership functions

   b. By using rule editor, rules are designed appropriately.

Figures 1b, 2b, 3b, 4b are inputs to the mamdani fuzzy classifier for truthness. Fig 1b shows membership function for Actors. Fig 2b shows membership function for UC (use case). Fig 3b shows membership function for CF (complexity factor). Fig 4b shows membership function for Application type.

Fig 5b show the membership function for output variable i.e. Real effort in person hours.

RMSE value is calculated for truth component i.e. 0.235.

1. By using the available sets, indeterminate inference system for indeterminate component is as given:

2. There should not be any overlapping between any two membership functions. Indeterminate and falsity membership functions lie only in those ranges which would have covered by two adjacent membership functions. The common area of membership function in fuzzy logic associated with indeterminacy and falsity.[5]

3. Using rule editor, appropriate rules are designed.

4. For each instance, final result is achieved by combining the results of all three components i.e. truth,

In this phase, truth value each data set is calculated. After that, Root mean square error (RMSE) is calculated for truth component in neutrosophic.

Figures 1c, 2c, 3c, 4c are inputs to the mamdani fuzzy classifier for indeterminacy. Fig 1c shows membership function for Actors. Fig 2c shows membership function for UC (use case). Fig 3c shows membership function for CF (complexity factor). Fig 4c shows membership function for Application type.

RMSE value is calculated for indeterminate component i.e. 0.2057.

<table>
<thead>
<tr>
<th>Fuzzy estimation result in RMSE</th>
<th>Neutrosophic estimation result in RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.253</td>
<td>(0.235, 0.206, 0.206)</td>
</tr>
</tbody>
</table>

Performance Comparison between fuzzy logic and neutrosophic logic classifier

This above table shows comparison between fuzzy and neutrosophic logic. Effort estimation of fuzzy logic is approximate 25% in RMSE. Effort estimation of neutrosophic logic is approximate 23.5% truth in RMSE, approximate 20.57% indeterminate in RMSE and approximate 20.57% false in RMSE.

**VII. CONCLUSION**

This paper uses neutrosophic logic for effort estimation for software applications. Previously used approaches are discussed and their disadvantages where they lack to estimate the efforts. Fuzzy logic approach is good it is not capable to deal with overlapping area of membership function. Due to this problem, there is some uncertainty and ambiguity in the information is arises. A new extended version of fuzzy logic is neutrosophic logic which supports to deal with overlapping area problem. This approach is applied in effort estimation of software applications. This approach gives the better result than previously used approaches.

**REFERENCES**

Neutrosophic Logic for Software Effort Estimation: An Improvement over Fuzzy Logic


15. A.Q. Ansari, Swati Aggarwal and Ranjit Biswas, “Neutrosophic Modeling and Control”, RCCCT (International conf. on computer and communication technology), 2010


